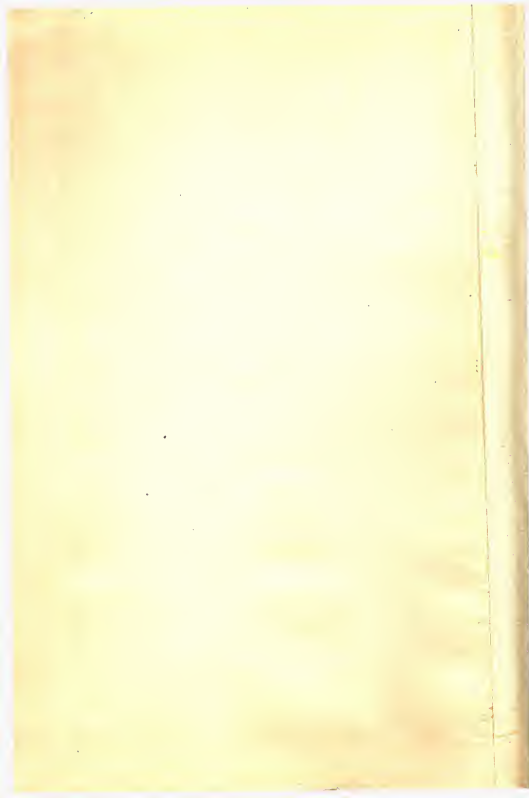


**СОЗДАНИЕ
И ИСПОЛЬЗОВАНИЕ
БАНКОВ ДАННЫХ В
АСУП**

Miller



СОЗДАНИЕ
И ИСПОЛЬЗОВАНИЕ
БАНКОВ ДАННЫХ
В АСУП



МОСКВА «СТАТИСТИКА» 1977

А. А. АРТЮКЕВИЧ, В. В. БАТЕЧКО, Е. И. ВЕЛЕСЬКО,
В. Е. ЦВИРКО

С58 **Создание и использование банков данных в АСУП.**
М., «Статистика», 1977.

175 с. с ил.

На обороте тит. л. авт.: А. А. Артюкевич, В. В. Батечко, Е. И. Велесько, В. Е. Цвирко.

Рассматриваются вопросы создания информационной базы АСУП на основе использования ЭВМ третьего поколения. Излагаются основные принципы построения, методы организации, контроля, хранения и актуализации банков данных АСУП. С учетом применения систем программирования БАСТАЙ и САВИ описываются состав и структура информационной базы АСУ Первого московского часового завода и отдельных задач, решаемых на ее основе.

Книга рассчитана на сотрудников вычислительных центров и проектно-конструкторских бюро, научно-исследовательских и проектных институтов, занимающихся разработкой информационного и программного обеспечения АСУ.

С 30502-153
008(01)-77 99-77

6Ф7.3

ПРЕДИСЛОВИЕ

Одним из самых ответственных моментов проектирования автоматизированной системы управления предприятием (АСУП) был и остается этап проектирования информационной базы и в первую очередь нормативно-справочной информации (НСИ), которая охватывает совокупность сведений, отражающих устойчивые свойства объектов и явлений производственно-экономической деятельности предприятия.

В АСУП, созданных на базе ЭВМ второго поколения, фонд НСИ на машинных носителях представлял собой набор массивов данных, ориентированных на использование в одной задаче или в заранее определенной группе задач. Для употребления этих же данных другими задачами массивы обычно переупорядочивались или создавались новые, ориентированные на другое применение. Такой подход становится неэффективным при возрастании объема обрабатываемых данных со сложными внутренними взаимосвязями. Можно без преувеличения сказать, что от организации и ведения информационной базы АСУП в настоящее время во многом зависит эффективность функционирования системы, продолжительность и трудоемкость ее разработки, поскольку на всех стадиях проектирования АСУП существует большое разнообразие взаимосвязей между нормативным хозяйством и задачами АСУП.

Одним из направлений эффективного решения названной проблемы является создание банков данных, что приводит к повышению актуальности данных, более строгой их согласованности, увеличению скорости выборки, надежности хранения и обеспечению независимости организации данных от проблемных программ.

Создание банков данных обуславливает принципиально новый подход к организации информационной базы и предполагает использование специальной системы управления базой данных (СУБД). Назначение, возможности и особенности применения одной из них описаны в первой главе. В системе БАСТАИ фактически отражены наиболее общие принципы, присущие многим СУБД, поэтому содержание первой главы позволяет читателю

получить представление о проблеме создания банков данных в АСУП при применении любой другой СУБД

Характерная особенность информационной базы АСУП — наличие ряда нормативно-справочных данных, которые целесообразно создавать и вести в виде независимых массивов. Опыт разработки и внедрения ряда систем показал, что в качестве программного средства для выполнения этой работы достаточно удобной является система САВИ. Она описана во второй главе.

Ряд специфических особенностей конкретного применения банков данных при разработке программ решения задач освещаются в третьей главе на примере Первого московского часового завода. Здесь приводится описание организации информационной базы и алгоритмов наиболее типичных задач АСУП.

ГЛАВА I

ОРГАНИЗАЦИЯ И ВЕДЕНИЕ БАНКОВ ДАННЫХ В АСУП

§ 1. Роль и место банка данных в АСУП

Совершенствование управления предприятием, а тем самым повышение производительности его работы обеспечивается широким внедрением автоматизированных систем управления предприятием (АСУП). АСУП рассматривается как взаимоувязанная совокупность технических, информационно-программных и организационных средств, направленных на выработку эффективных научно обоснованных решений по управлению ресурсами и производственной деятельностью предприятия. При этом автоматизация обработки информации с использованием ЭВМ является средством совершенствования методологии управления, играет важную роль, хотя и вспомогательную, в решении проблемы совершенствования планирования и управления предприятием.

Автоматизация обработки данных с помощью ЭВМ требует в свою очередь построения рациональных потоков информации на предприятии, формализации ее представления, создания специальных методов организации и обработки данных. До внедрения ЭВМ весь информационный процесс расчленялся на отдельные, зачастую дублирующие друг друга части, обусловленные организационной структурой предприятия, в результате нарушалась преемственность потоков информации и снижалась достоверность получаемых данных. Применение ЭВМ обеспечивает синтез информационных потоков, создание единой информационной базы и многоцелевое ее использование в процессе решения функциональных задач (рис. 1).

Состав данных информационной базы определяется составом и содержанием задач, решаемых в вычислительном центре (ВЦ). Число задач и их содержание постоянно меняются, а соответственно изменяются информационная база и взаимосвязь между задачами, решаемыми в ВЦ и службах предприятия. Обработка

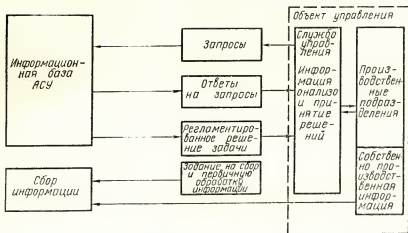


Рис. 1. Схема связей информационной базы АСУ и объекта управления

информации на ЭВМ находится в тесной связи с обработкой данных в целом на предприятии, поэтому создание АСУП не нарушает единства информационного процесса на предприятии, способствует его качественному улучшению. Это выражается в своевременном обеспечении управленческого персонала достоверной и актуальной информацией. Для хранения и накопления ее требуются запоминающие устройства ЭВМ большой емкости.

В разрабатываемых до сих пор АСУП в качестве таких устройств в основном использовались магнитные ленты (МЛ), применялся последовательный метод организации информации в виде независимых массивов на МЛ, что наложило отпечаток на организацию системы обработки данных. Как правило, при решении конкретной задачи в традиционных АСУП требуется непосредственное участие в расчетах не всей нормативно-справочной информации, а лишь относительно небольшой части, а для поиска ее в массиве с последовательной организацией данных приходится просматривать весь массив. По этой причине длительность решения задачи зависит в основном от объема просматриваемой информации, а не от непосредственно участвующей в решении задачи.

С появлением устройств прямого доступа большой емкости — магнитных дисков — появилась возможность накопления больших объемов информации существенно новым способом — в виде банка данных.

Банк данных — это система организации, ведения и хранения интегрированной информации, расположенной на машинных носителях и предназначенной для комплексного многоцелевого использования, вместе со специальными программами, организационными и техническими средствами его ведения. Основные состав-

ные элементы банка данных — база данных (БД) и система управления базой данных (СУБД).

База данных представляет собой взаимосвязанную совокупность данных определенным образом организованных, хранимых на магнитных дисках и им подобных запоминающих устройствах прямого доступа и используемых в качестве исходной информации для решения задач. База данных может иметь сложную логическую структуру взаимосвязи данных (сетевую, древовидную и др.).

Структура базы данных должна удовлетворять требованиям многих применений, обеспечивая быстрый доступ к каждому элементу информации. Для этого в БД должны быть отражены естественные логические связи между данными, на основании которых при решении задач может выполняться выборка логически связанных записей без обработки остальной информации, находящейся в базе данных. Такой целенаправленный поиск информации сокращает время решения задач, алгоритмы которых не позволяют осуществить выборку необходимых данных последовательным просмотром массива без предварительной его сортировки или объем информации которых незначителен по сравнению с общим объемом массивов.

В базе данных отсутствует избыточная информация, т. е. значение каждого реквизита (например, номер детали, код материала) запоминается в базе данных только один раз, а поиск нужных данных (например, применяемости детали в узле) по соответствующему реквизиту осуществляется с помощью адресных ссылок к месту ее расположения в базе данных. Это приводит к экономии оперативной памяти, используемой для размещения информации в процессе ее обработки, но не всегда обуславливает экономию памяти, предназначенной для хранения информации в БД. Последнее зависит от разности между объемом памяти, занятой адресными ссылками, и объемом данных, которые они заменяют.

В базе данных повышается достоверность информации вследствие программной проверки согласованности различных данных между собой. При добавлении в базу данных новых единиц информации и при обновлении ее этот процесс будет успешным только в том случае, если изменение не противоречит состоянию ранее накопленных данных. Для этого вновь включаемые единицы информации вплетаются в ранее построенную сеть адресных связей и становятся доступными для обработки. Например, нельзя включить в БД данные конструкторской спецификации по тем деталям и узлам, по которым нет информации в массиве предметов, содержащем сведения о соответствующих деталях и узлах.

Система управления базами данных (иногда ее называют операционной системой банка данных) включает комплекс программ для ввода данных, загрузки и корректировки базы данных, обеспечения доступа к данным из прикладных программ или с терминалов. СУБД обеспечивает сохранность, восстановление базы данных и контроль за ее использованием. Как правило,

СУБД разрабатывается в виде общесистемных пакетов прикладных программ и предназначается для многих применений. Известные в настоящее время СУБД (БАСТАИ, САВИ, СИОД1, СИОД2, УНИБАД, СИНБАД, ОКА, БАНК и др.) различаются структурой БД, характером выполнения ими своих функций. Так, в системе БАСТАИ предусмотрен доступ к данным только на уровне Ассемблера, в СИОД1 — дополнительно на Коболе и ПЛ/1. В то же время БАСТАИ имеет программные компоненты, обеспечивающие создание специальных программ подготовки данных, а в СИОД1 они отсутствуют. СИОД2 — модифицированный и несколько расширенный вариант СИОД1 и программно с ним совместим. В принципе эти три СУБД ориентированы на жесткий состав и структуру массивов базы данных и на использование в АСУП для организации и ведения нормативно-справочной информации.

Система БАНК представляет собой комплекс программных средств для организации информационной базы различной структуры. В этом состоит преимущество данной СУБД по сравнению с СИОД1 и СИОД2 [7]. Однако из-за применения в системе БАНК [5] метода рандомизации неоправданно увеличивается расход памяти, что резко ограничивает его применение в АСУП.

Секторно-ориентированная система программирования САВИ — универсальная система организации и обслуживания независимых массивов информации на машинных носителях. Данная система как самостоятельно, так и в сочетании с СУБД БАСТАИ или СИОД1 и СИОД2 — достаточно удобное средство организации и ведения нормативно-справочной информации. СИНБАД и ОКА по своим характеристикам почти тождественны. В них предусмотрена организация и ведение баз данных иерархической структуры.

С применением банков данных изменилась структура систем обработки данных. Это отличие в первую очередь заключается в том, что процедуры обработки информации (ввод, корректировка, сортировка, дублирование и контрольные распечатки массивов) не относятся к конкретным задачам, а выделены в самостоятельную систему — часть СУБД. Второй особенностью является функционирование программ в динамической среде вычислительных ресурсов, что обуславливается наличием средств настройки программ обработки на параметры вычислительной системы, а также возможностью параллельного решения задач и их синхронизации. Третья особенность новой системы обработки — автоматическая подготовка программ к работе на основании параметров, записанных на специальном языке. И последняя особенность — отделение описания структур данных от программ обработки.

В основу разработки СУБД положен принцип независимости данных, что означает отделение функций ввода, хранения и корректировки данных от программ обработки, и независимости прикладных программ от изменения физических характеристик и размещения данных, а также от наличия неиспользуемых типов данных. Независимость достигается хранением описания данных

отдельно от программы. Объединять программу с описаниями данных можно во время выполнения программы или на этапе компиляции. В СУБД часто сочетаются оба метода: на этапе компиляции осуществляют привязку на уровне структуры записей, а во время выполнения программы происходит привязка на уровне размещения записей.

Независимость данных позволяет изменять их описание без последующего изменения программ, использующих эту информацию. К тому же программиста на этапе написания программы не интересует взаимосвязь между используемой и неиспользуемой им из БД информацией. Взаимосвязь между данными программист учитывает лишь на этапе составления алгоритма решения задачи, так как это в некоторой степени влияет на время решения задачи. Указанные факторы определяют направление проектирования систем электронной обработки данных в АСУП, их реализация обуславливает надежность и простоту применения системы, несмотря на ее внутреннюю сложность.

§ 2. Назначение и область применения системы БАСТАЙ

БАСТАЙ* представляет собой проблемно-ориентированную систему программирования по организации и ведению банка данных, включая загрузку данных, их корректировку и реорганизацию. В системе имеются программные компоненты для поиска информации в банке данных при разработке программ пользователя. Являясь, по существу, универсальным генератором программ для создания банков данных, БАСТАЙ вместе с тем ограничена по своим возможностям: массивы банка данных могут включать основные и наиболее взаимосвязанные данные фонда НСИ (сведения о составе изделия, технологических маршрутов, оборудования, материалах и др.), но не в состоянии одновременно содержать информацию всех нормативно-справочных документов (НСД), которые подвергаются автоматизированной обработке при создании АСУП. Кроме того, в процессе эксплуатации АСУП иногда требуется создать отдельные массивы для корректировки и разработки новых задач.

Массивы информации, не включенные в банк данных, могут быть организованы с помощью оригинального математического обеспечения либо с помощью системы САВИ** (см. гл. 2).

В зависимости от состава и содержания недостающая информация может быть включена в другой банк данных, организованный с помощью той же системы БАСТАЙ. Следует лишь учитывать, что чем больше исходных НСД будет включено в банк данных, чем сложнее взаимосвязи между ними, тем труднее на начальной

* BASTEI — Bankspeicherung Technischer Informationen (банк накопления технической информации) [10].

** SAWI — Speichern und automatisches Wiederauffinden (накопление и автоматический поиск информации) [10].

стадии проектирования и внедрения решаются вопросы организации банка данных, его корректировки и реорганизации. Не лишним будет также отметить, что чем сложнее структура массивов, тем больше трудностей может возникнуть при восстановлении банка данных, если в процессе работы с ним или при его корректировке был допущен брак в работе оператора, брак при подготовке данных или несвоевременно выявлены отказы в работе ЭВМ. Учитывая это, разработчики структуры массивов банка данных и математического обеспечения должны внимательно относиться к вопросу определения структуры массивов и в каждом конкретном случае оценивать возможность появления названных выше трудностей.

Несмотря на то что система БАСТАИ по замыслу авторов была ориентирована на решение задач подсистемы технической подготовки производства, ее можно использовать и при решении других задач в других подсистемах, если исходные данные определенным образом взаимосвязаны между собой. В частности, данная система может быть использована при организации базы данных для оперативного планирования, в торговле, в системе образования и др. Так, в торговле сведения о товарах и их поставщиках могут быть представлены в виде базовых массивов, а сведения о цене, количестве, сроках поставки и т. п. составят массив связей.

В системе образования базовые массивы могут включать сведения о возможностях использования преподавателя для одного или нескольких предметов различного уровня преподавания. Массив связей должен отражать взаимосвязи между массивами. На основании указанных сведений можно получить ответы на запросы относительно замены преподавателей, перечня мероприятий, которые может провести каждый преподаватель, и т. д.

Конкретные программы создаются путем генерации на основании параметров, описывающих требуемую структуру массива банка данных или требуемых программ поиска. Программирование параметров может выполняться самим потребителем системы.

Программа поиска составляется на языке Ассемблера с использованием макрокоманд систем БАСТАИ и ДОС ЕС и позволяет обрабатывать базовые данные последовательным методом, по непосредственным адресам, по конкретным значениям ключевых реквизитов или интервалу значений, осуществлять обработку накопленных цепей, выборочную обработку с помощью создания временных цепей и др. Макрокоманды более высокого уровня позволяют получать табуляграммы путем задания параметров о структурном составе или технологии производства.

Полученные в результате генерации программы настроены на конкретную структуру информационной базы. После изменения структуры банка данных генерацию необходимо провести заново, уточнив соответствующим образом параметры. При реорганизации базы данных не только приводятся в соответствие логическая и физическая последовательности записей в массивах, но и выполняются добавление, корректировка и стирание данных.

Таким образом, организованные массивы системы БАСТАЙ не только обеспечивают актуальность информации, минимальную избыточность ее, но и позволяют эффективно решать вопросы поиска данных, экономить затраты на программирование и отладку программ задач АСУП. По предварительным оценкам, затраты на приспособление системы БАСТАЙ к условиям пользователя составляют 5—10% от затрат, требующихся при обычной разработке программ организации и ведения банка данных.

БАСТАЙ работает под управлением дисковой операционной системы ДОС ЕС и требует наличия накопителей на магнитных дисках (МД) и магнитных лентах (МЛ). Для работы системы необходимы также: центральный процессор; оперативное запоминающее устройство емкостью не менее 32 тыс. байт; устройство ввода с перфокарт; устройство связи оператора с ЭВМ (пишущая машинка); алфавитно-цифровое печатающее устройство.

Количество накопителей на МД определяется пользователем в зависимости от объема включаемых в банк данных, однако не может превышать 16 (минимальное количество 2). На МЛ используется не более четырех накопителей, минимум — один. Никаких других ограничений на применение системы не накладывается.

§ 3. Состав и структура массивов системы БАСТАЙ

С помощью системы БАСТАЙ можно создать от одного до четырех массивов данных, включая массивы, содержащие их взаимосвязи.

Среди массивов банка данных следует различать базовые массивы и массивы связей. Базовые массивы — это массивы, каждая запись которых содержит информацию о рассматриваемом объекте. Массивами связей являются массивы, содержащие структурные связи между записями одного массива или связи между базовыми массивами и данные, характеризующие эти связи. На рис. 2 стрелками указаны массивы входов в банк данных при обращении к

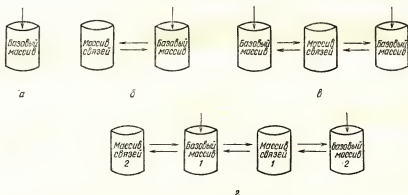


Рис. 2. Варианты организации массивов в системе БАСТАЙ

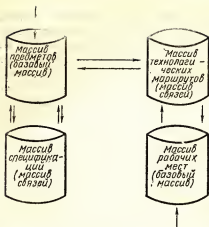


Рис. 3. Вариант структуры базы данных для машино- и приборостроительных организаций

ним и существующие между массивами взаимосвязи. Варианты *а* и *б* содержат по одному базовому массиву, а варианты *в* и *г* — по два. По одному массиву связей содержат варианты *б* и *в*, а вариант *г* дополнительно включает второй массив связей, в котором зафиксированы структурные связи между записями одного из двух базовых массивов.

Предлагаемая для машино- и приборостроительных предприятий структура базы данных (рис. 3) в основном определяется тем, что она предназначена для хранения и обработки конструкторско-технологической информации. Массив

предметов (МП) и массив рабочих мест (РМ) являются базовыми, два других — массивами связей. В МП, как правило, хранятся данные об изделиях, их составных частях и материалах. Каждому предмету (изделию или материалу) соответствует одна запись, которая содержит характеристики данного предмета, не зависящие от его связи с другими предметами.

Связь между отдельными предметами осуществляется посредством адресных ссылок через записи массива спецификаций (СП). Каждая из них соединяет пару записей в МП и содержит данные, характерные только для этой связи, т. е. такие данные, которые не могут быть отнесены к каждому предмету в отдельности (например, количество деталей в некотором изделии или количество материала на деталь).

РМ содержит сведения о наличии оборудования и его характеристики. С помощью записей технологических операций, содержащихся в массиве технологических маршрутов (ТМ), осуществляется связь между предметами и оборудованием. На линии этой связи, т. е. в записях ТМ, располагаются данные, конкретизирующие выполнение операций для данной детали на данном оборудовании. Обработка информации начинается всегда с записи, содержащейся в базовом массиве, — МП или РМ. По содержащемуся в базовой записи адресу можно выбрать запись связи, а при необходимости и соответствующую запись подчиненного базового массива. С помощью последующих записей массивов МП или РМ поиск можно продолжить. Характерно, что при таком подходе будут выбираться только логически связанные записи, вся остальная информация, находящаяся в банке данных, в поиске не участвует, что заметно сокращает время поиска и обработки требуемой информации.

В основу банка данных системы БАСТАИ положен принцип цепной адресации. В каждой базовой записи содержатся начальные адреса цепей, соединяющих логически связанные записи массивов. В массиве предметов могут брать свое начало четыре цепи: цепь состава изделий (цепь спецификаций), цепь применения предметов, цепь рабочих операций, обратная цепь рабочих операций. Первые две цепи обязательные, необходимость построения двух других цепей задается с помощью параметров при генерации системы. Дополнительно в массиве предметов можно предусмотреть создание рабочей цепи. В отличие от вышеперечисленных цепей она создается не программами загрузки или изменения банка данных, а программами обработки. Эту цепь целесообразно строить в случае, когда требуется многократно обрабатывать некоторое множество записей, выбранных по определенному критерию (критериями могут быть перечень деталей, поступающих по кооперации, перечень изделий, изготавливаемых на экспорт, и т. д.). Подлежащие обработке записи сначала соединяются в рабочую цепь, а затем подвергаются обработке. Создание рабочих цепей позволяет сократить время поиска информации, особенно в случае, когда объем ее мал по сравнению с объемом данных, накопленных в банке данных. В других случаях использование рабочих цепей может значительно упростить алгоритм расчета.

В РМ может начинаться только цепь использования рабочих мест. Необходимость построения этой цепи задается с помощью соответствующих параметров.

Ценная адресация записей банка данных возможна благодаря наличию в записях массивов системной части, в которую программы загрузки и обслуживания банка данных помещают прямые физические адреса записей, включаемых в соответствующие цепи. Адреса состоят из номера цилиндра, номера дорожки и номера записи на дорожке. Для экономии места на диске адреса представлены в сжатом формате; их длина не более четырех байт. Помимо адресов в системной части записей могут располагаться и другие поля, предусмотренные в системе: поля для счетчиков записей в цепи и контрольные поля. Первые резервируются только при задании специальных параметров генерации. При этом счетчики содержат число записей в соответствующей цепи. Контрольные поля располагаются в системной части записей массивов связей и предназначены для дополнительного контроля достоверности информации в банке данных.

В контрольном поле при загрузке или добавлении записей помещается часть кода предмета или рабочего места. При считывании записи связи содержимое контрольного поля сравнивается с соответствующей частью кода предмета или рабочего места и в случае их несовпадения выдается сообщение об ошибке. Длина контрольного поля и положение части кода, выбираемой для сравнения с контрольным полем, задаются параметрами при генерации системы.

Для последовательной обработки базовых массивов в системной части записей содержится адрес следующей записи массива. Необходимость этого адреса обусловлена специфической организацией базовых массивов. Для загрузки базовых массивов требуется, чтобы входные записи были рассортированы в порядке возрастания кодов предметов или рабочих мест. После загрузки массива физический порядок расположения записей на диске соответствует их логическому порядку, и адрес следующей записи указывает на следующую физическую запись. При добавлении новых записей они записываются вслед за последней записью массива, если значение кода добавляемой записи больше значения кода последней записи массива, или же записываются в области переполнения, если это не так. В последнем случае физический порядок записей не соответствует логическому, поэтому адрес в записи, за которой должна стоять добавляемая, указывает на добавленную запись в области переполнения, а адрес в добавленной записи — на следующую за ней в логическом порядке запись в основной области данных.

В системной части записей массива предметов предусмотрено двухбайтовое поле для текущего номера состояния массива. Наличие этого поля позволяет эффективно продолжить обработку данных при аварийном прерывании обрабатывающей программы (например, из-за отказа ЭВМ). В каждом массиве банка данных имеется специальная управляющая запись с характеристикой массива. В управляющей записи МП содержится также и текущий номер, который увеличивается на единицу при открытии массива. В процессе обработки каждой записи массива предметов перед помещением ее обратно на диск с помощью специальной макрокоманды можно приписать текущий номер из управляющей записи. Если произошло прерывание программы и массив не закрыт (т. е. обработка не закончена), управляющая запись в массив не записывается. При повторном выполнении программы текущий номер в обработанных записях будет равен текущему номеру в управляющей записи, поэтому нет необходимости повторять обработку этих записей, что в конечном счете позволит сократить затраты машинного времени на повторное выполнение программы.

Двухбайтовое поле для ступени диспозиции является вспомогательным средством для реализации эффективных алгоритмов расчетов, связанных с определением состава и структуры изделий. Ступень диспозиции рассчитывается и корректируется программами загрузки и изменения массива спецификаций и принимает значение, равное максимальной ступени вхождения предмета во все изделия, включенные в банк данных (для «конечных» изделий значение ступени диспозиции равно нулю).

Помимо перечисленных выше цепей в банке данных системы БАСТАЙ можно предусмотреть наличие еще двух дополнительных цепей: обратной цепи применения предметов и обратной цепи использования рабочих мест. Полная структура системных частей

записей массивов банка данных и порядок расположения отдельных полей в записях приведены в приложении 1 (табл. 1—4).

В то время как структура системных частей записей полностью определена назначением и возможностями системы БАСТАИ, на структуру и содержание части пользователя в записях система не накладывает практически никаких ограничений. В часть пользователя могут включаться любые данные, вводимые с первичных нормативно-справочных документов, а также данные, получаемые в процессе обработки. Если в массиве несколько типов записей, то в часть пользователя можно включить признак типа записи. Структура части пользователя для разных типов записей может быть различной, однако все записи в пределах одного массива должны иметь фиксированную (одинаковую) длину. Это единственное требование, которое необходимо учитывать при определении структуры записей массивов банка данных.

В некоторых случаях обозначение типа записи для базовых массивов удобнее включать не в часть пользователя, а в системную часть в качестве старших разрядов кода предмета или рабочего места. В этом случае записи одного типа оказываются расположенными рядом, что особенно выгодно при последовательной обработке. Такая ситуация может возникнуть, например, при включении в массив предметов не только изделий и их составных частей, но и материалов. Если дополнительно для каждого типа записей ввести в базовый массив искусственные записи с минимальным для данного типа кодом, это даст возможность перейти к последовательной обработке нужного типа записей непосредственно, без рассмотрения записей других типов.

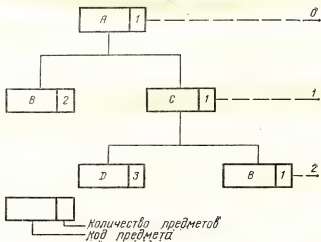


Рис. 4. Структура изделия А

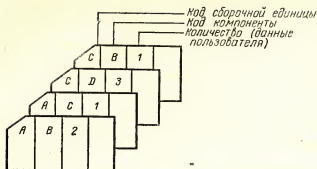


Рис. 5. Спецификация на изделие А

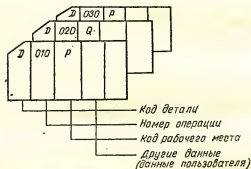


Рис. 6. Маршрут обработки детали Д

Наглядное представление записей массивов банка данных и связей между ними дают рис. 4—7.

§ 4. Создание программ организации и обслуживания банка данных

Программное обеспечение системы БАСТАИ состоит из обобщенной проблемной программы (ОПП), генератора и библиотек исходных и объектных модулей. Из ОПП на основании задаваемых параметров генерируется ряд модулей, которые после трансляции и редактирования объединяются в программы для выполнения различных функций по созданию и обслуживанию банка данных. Помимо параметров необходимо написать также ряд специальных подпрограмм пользователя (СПП), предназначенных для обработки входных записей при включении их в банк данных и для распечатки загружаемых или изменяемых записей.

Программы создания и обслуживания банка данных состоят из

управляющего модуля LEITPROG, модулей организации массивов DATEIORG и модулей ввода-вывода EAWURZEL, EAARBEIT и EAEROABS. Если в систему обслуживания включаются программы реорганизации, дополнительно генерируются модули BDAREORG и VDAREORG, выполняющие в программах реорганизации функции управляющего модуля. Требуемые комбинации модулей для выполнения различных функций представлены в табл. 1.

Таблица 1

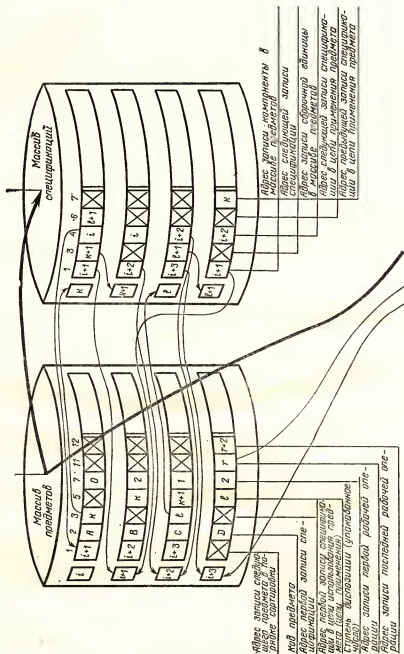
Имя модуля	Загрузка массивов				Изменение всех массивов	Реорганизация массивов			
	МП	РМ	МС	ТМ		МП	РМ	МС	ТМ
LEITPROG	+	+	+	+	+	—	—	—	—
BDAREORG	—	—	—	—	—	+	+	—	—
VDAREORG	—	—	—	—	—	—	—	+	+
DATEIORG*	+	+	+	+	—	+	+	—	—
DATEIORG**	—	—	—	—	+	—	—	—	—
EAWURZEL	—	—	+	+	+	+	+	+	+
EAARBEIT	—	—	+	+	+	+	+	+	+
EAEROABS	—	—	+	+	+	+	+	+	+

* Для загрузки каждого массива требуется генерировать отдельный модуль.

** Для изменения всех массивов генерируется один модуль.

Перечень параметров, их назначение и порядок следования приведены в приложении 2. С помощью параметров описывается структура входных записей, структура записей в банке данных, относительные адреса расположения реквизитов и системных полей в записях, их длина, положение ключей сортировки, задаются точки включения СПП, определяются требуемые адресные цепи. Некоторые параметры зависят от выбранной структуры банка данных и могут опускаться.

Параметры записываются на бланках Ассемблера и перфорируются на картах, начиная с колонки 1, операнды параметров — начиная с колонки 16, комментарии — с колонки 25. Порядок следования параметров должен соответствовать порядку их расположения в таблицах (см. приложение 2). Структура входных записей и записей на диске описывается на языке Ассемблера с помощью инструкций DS, DC и ORG. Описание структуры записей для каждого массива удобнее всего помещать в библиотеку исходных модулей в виде отдельных книг. При программировании параметров в этом случае вместо описания структуры записей достаточно включить оператор Ассемблера COPY. Эти же описания нужно использовать в программах подготовки исходных данных и в обрабатывающих программах. Структуру записей на диске можно представить либо в виде одной книги, включающей структуру системной части и части пользователя, либо отдельно. Если структура представлена в виде одной книги, имена полей в системной части должны отличаться от имен, приведенных в



приложении 1. В этом случае при программировании параметров после параметров DFSATZGD, DFSATZSD, DFSATZMD и DFSATZAD необходимо задавать соответственно операторы Ассемблера ORG GD#SATZ, ORG SD#SATZ, ORG MD#SATZ и ORG AD#SATZ, а вслед за ними оператор COPY и имя требуемой книги. Если описание структуры записи закаталогизировано в виде двух книг, в операторе COPY указывается имя книги, содержащей описание структуры части пользователя, оператор ORG в этом случае задавать не нужно.

После программирования параметров и СПП можно приступить к генерации системы. Управляющие карты для выполнения генерации приведены в приложении 3. В одном задании можно генерировать сразу несколько модулей. Признаком начала параметров для одного модуля является параметр PROGRART, признаком конца этого модуля — параметр PROGRART для следующего модуля. Признаком конца задания будет наклонная черта и звездочка в колонках 1 и 2. Ввод параметров производится с символического устройства SYSIPT, вывод сгенерированных модулей осуществляется на магнитную ленту. Модули отделяются на ленте друг от друга ленточными марками. Если в процессе генерации обнаружены ошибки, ошибочный модуль на МЛ не выводится. Полученные в процессе генерации модули представляют собой исходные программы на языке Ассемблера, поэтому их необходимо оттранслировать и поместить в библиотеку объектных модулей.

§ 5. Подготовка исходной информации для загрузки и изменения банка данных

Программы загрузки и изменения банка данных позволяют вводить исходную информацию с перфокарт или с магнитных лент. Для ввода с перфокарт используется символическое устройство SYSIPT, для ввода с магнитной ленты — SYSØ10. При больших объемах информации в качестве устройства ввода предпочтительно применять магнитную ленту. Программы системы БАСТАИ предполагают, что входные массивы на магнитных лентах состоят из блокированных записей переменной длины (VARBLK). Для загрузки банка данных требуется по одному входному массиву для каждого массива банка данных, для изменения всех массивов банка данных достаточно иметь один входной массив. Входная информация для загрузки должна быть соответствующим образом рассортирована в порядке возрастания ключевых реквизитов. С целью экономии времени на внесение изменений входные данные также лучше рассортировать.

Наиболее простым и наименее трудоемким является следующий подход к разработке программ создания и обслуживания банка данных. С помощью специальной программы строятся фиктивные массивы для загрузки банка данных, а вся действительная информация включается в банк данных в режиме внесения изменений.

Фиктивный массив для загрузки массива предметов состоит из двух записей, имеющих минимально допустимые значения кода предмета (например, если код предмета должен быть упакован, в качестве минимальных ключей можно выбрать упакованные нуль и единицу). В остальных входных массивах достаточно иметь по одной записи с минимальными значениями ключей, при этом в записях связей для МС и ТМ должны использоваться те же коды предметов и рабочих мест, что и для базовых массивов. Во всех входных записях на одном и том же месте должен быть признак функции загрузки для соответствующего массива. Место расположения признака функции во входных записях задается для модуля LEITPROG параметром FKZEICHN, а значение признака — параметрами GDFKLADN, SDFKLADN, MDFKLADN и ADFKLADN.

Входная информация для изменения банка может быть подготовлена на перфокартах или на перфолентах. Так как изменения могут поступать сразу для нескольких входных документов различной структуры, каждому документу необходимо разработать макет перфорации. Кроме макетов перфорации, во входных записях должен содержаться также признак функции изменения. В системе предусмотрено выполнение следующих функций:

- добавление новых записей;
- удаление записей из массива (стирание);
- корректировка записей (замена неведущих реквизитов);
- групповое стирание записей (удаление всех записей спецификации на сборочную единицу или записей всех рабочих операций для некоторой детали).

Для массивов связей допускается замена. В МС с помощью этой функции можно заменить компоненту и признаки сортировки, в ТМ — рабочее место и признаки сортировки. Во входных записях в этом случае должны содержаться старые и новые значения заменяемых реквизитов. В массивах РМ и ТМ система может выполнять отделение и соединение, что означает разъединение или установление связей между записями рабочих операций в ТМ и записями рабочих мест в РМ.

В зависимости от значений макета и признака функции в программе ввода и записи исходных данных на магнитную ленту нужно предусмотреть контроль входных данных на достоверность, корректность задания функций изменения, формирование записей для ввода в банк.

В зависимости от заданной во входной записи функции изменения программа должна проверять наличие в банке данных записей, подлежащих изменению, подготавливать недостающие в базовых массивах записи, если необходимо добавлять новые записи связей, а требуемые базовые записи в банке данных отсутствуют. Все ошибочные записи не должны включаться в выходной массив; в программе следует предусмотреть их распечатку. После работы программы записи для изменений должны быть рассортированы в следующем порядке: записи для добавления и коррек-

тировки базовых массивов; записи для изменения массивов связей; записи для стирания в базовых массивах.

Чтобы обеспечить такой порядок рассортированности, необходимо выбрать признаки функций изменения и структуру записей для каждого массива. Чтобы разделить записи, относящиеся к разным массивам, достаточно ввести во все записи в качестве старшего ключа сортировки признак принадлежности записей к массивам банка данных. Признаками могут быть: 1 — для записей на добавление и корректировку МП, 2 — для записей на добавление и корректировку РМ, 3 — для записей на изменение МС, 4 — для записей на изменение ТМ, 5 — для записей на стирание в МП и 6 — для записей на стирание в РМ. В качестве второго ключа сортировки нужно взять: код предмета для МП, код рабочего места для РМ, код сборочной единицы (куда входит) для МС и код детали для ТМ.

Для массивов связей требуется дополнительно еще от одного до трех признаков сортировки. Для МС в качестве дополнительного признака сортировки обычно выбирают код компоненты (что входит), для ТМ — номер рабочей операции и код рабочего места. В качестве последнего ключа сортировки для записей всех типов следует задавать признак функции изменения. В программе подготовки входных данных структуру подготовленных записей нужно определить таким образом, чтобы запись состояла из двух частей — области ключей и области остальных данных. Длина области ключей определяется как максимальная суммарная длина ключей сортировки для некоторого типа записи. Оставшиеся свободные байты в области ключей для других типов записей можно заполнить пробелами. Последний ключ сортировки — признак функции — нужно поместить в конце области ключей. Признак функции во входных записях для загрузки и изменения банка данных помимо требуемой функции должен указывать и на массив банка данных, к которому относится эта запись. Признак удобно представить двумя символами, из которых первый указывает на массив (например, 1 — МП, 2 — РМ, 3 — МС, 4 — ТМ), а второй — на функцию (например, 0 — загрузка, 1 — стирание, 2 — добавление, 3 — корректировка, 4 — стирание групповое, 5 — замена, 6 — отделение, 7 — соединение).

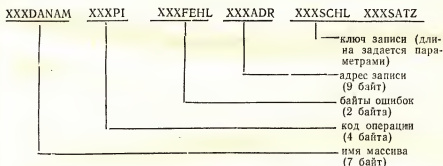
В подготовленном таким образом и рассортированном массиве могут встречаться одинаковые записи, так как изменения исходных документов подготавливаются различными службами предприятия и могут дублировать друг друга. При изменении банка данных воспринимается только первая из повторяющихся записей, остальные записи выводятся в протокол ошибок, выдаваемый на пишущую машинку пульта оператора, что может вызвать значительные потери машинного времени. Для сокращения их из рассортированного массива уделяют все повторяющиеся записи, например, путем перезаписи массива на другую магнитную ленту. Полученный в результате такой процедуры массив предназначен для непосредственного ввода в банк данных.

Примерный поток управляющих карт для изменения банка данных приведен в приложении 4.

§ 6. Макрокоманды системы БАСТАИ

Для обработки накапливаемых в банке данных в системе имеется ряд макрокоманд, позволяющих извлекать записи из массивов и помещать их обратно. В макрокомандах используются регистры 9, 10 и 12. Применение этих регистров вне макрокоманды разрешается только при запоминании их содержимого и восстановлении перед обращением к макрокомандам. Регистры 0, 1, 14 и 15 можно использовать в программах, однако их содержимое изменяется большинством макрокоманд ДОС и БАСТАИ. Из глобальных переменных в макрокомандах употребляются: &ADRU, &ADRUD, &ADRUS, &AGI3, &AGI4, &ANT, &SYRA1, &SYRA2 и &TE. Использовать эти переменные в своих программах не разрешается.

Записи, получаемые из банка данных с помощью макрокоманд, помещаются и обрабатываются в специально отведенных для каждого используемого массива областях. Область записи должна состоять из трех частей: префикса, системной части и части пользователя. Область префикса нужно определять не с помощью инструкции Ассемблера DS, а с помощью макрокоманды VA#RB, вслед за которой необходимо отвести место для записи (системной части и части пользователя). Структура префикса приведена ниже.



Поля префикса устанавливаются и используются как макрокомандами, так и программами. Все поля в префиксе имеют жестко закрепленные за ними имена и под этими именами их можно использовать в программах на языке Ассемблера. Имена полей состоят из приставки XXX и собственно имени. Приставка берется из макрокоманды VA#RB и служит для отличия одинаковых полей в префиксах для разных массивов. Поле для ключа записи имеет длину, равную длине кода предмета или рабочего места.

Макрокоманды присваивают байтам ошибок определенное значение (табл. 2).

Пятизначное значение байта ошибок	Причина ошибки
2000	Неустрашимая ошибка ввода-вывода
0400	Нарушена организация массива
0080	Ошибка данных в поле счетчика
0010	Сбой при чтении ключа или данных
0008	Запись не найдена
0004	Неверен код операции (чтения или записи)
0002	Конец массива

В табл. 3 приводится перечень макрокоманд системы БАСТАИ. В таблице приняты обозначения:

- ИМ — символическое имя поля;
 ИП — имя префикса (имя при макрокоманде VA#RB);
 DI, D2 ... — имена DTF для массивов ДЭС;
 X, X1, X2, ... — имена массивов банка данных (обязательно семь символов);
 ОП — код операции;
 К — количество;
 P1, P2, ... — номера регистров;
 П — приставка;
 DK — длина ключа;
 Т — текст (строка символов);
 А — адрес.

Таблица 3

Символическое имя поля	Макрокоманда	Операнды
[ИМ]	VA#RB	П[,X][,DK][,PSAB]
[ИМ]	BE#AD	ИП
[ИМ]	BE#SL	ИП
[ИМ]	LE#SA	ИП
[ИМ]	ZU#SA	ИП
	SE#QU	К[,ОП1][,ОП2][,ОП3]
	DA#NA	X, ИМ
[ИМ]	DE#RO	[(D1, D2, ..., D5)][, (ИП1, ИП2, ..., ИП4)]
[ИМ]	DA#BS	[(D1, D2, ..., D5)][, (ИП1, ИП2, ..., ИП4)]
[ИМ]	KE#TT	ANKADR=, FOLGADR=, BASADR=, [ARBERB=, ARBERV=,] ROUT=
[ИМ]	RU#PI	ОП, ИП[, ИМ]
[ИМ]	ST#AR	(P1, P2, ..., P6), А[, JA или NEIN]
[ИМ]	TE#XT	T[, A][, K]
[ИМ]	AD#RU	DEF=, DBER=, DRULNG=, EINHEIT=, ZEIL=, UEB=, LEERZ=, DATUM=, SEITE=, REG=

Рассмотрим функции перечисленных в табл. 3 макрокоманд.

VA#RB — резервировать область для префикса. Эта макрокоманда требуется для каждого используемого в программе массива. Первым операндом макрокоманды является приставка, которая представляет собой последовательность алфавитно-цифровых символов, начинающихся с буквы. Максимальная длина приставки три символа. Приставка является составной частью имен полей префикса, например поле ключа будет иметь имя MPSCHL, если приставка состоит из последовательности символов MP. Макрокоманда должна иметь имя. Оно присваивается первому байту префикса и имеет характеристику длины, равную единице. К префиксу также можно обращаться по имени первого поля префикса — MPDANAM (здесь MP — приставка).

Параметр PSAB задается в случае, если область записи определяется как фиктивная секция; имя массива и длину ключа DK в этом случае можно опускать. Длина ключа обязательно задается для базовых массивов и для MC (берется длина ключа в записях MP). Для TM длину ключа задавать не разрешается.

BE#AD — читать базовую запись по адресу. В операнде этой макрокоманды задается имя префикса. Адрес записи на диске перед обращением к макрокоманде необходимо поместить в префиксе в поле XXXADR. Пересылку адреса в префикс нужно выполнять с помощью макрокоманды RU#RI. Дополнительно в программе должна быть задана макрокоманда SE#QU.

BE#SL — читать базовую запись по ключу. Значение ключа (код предмета или рабочего места) перед применением макрокоманды помещают в поле ключа в префиксе (например, по команде MVC). Дополнительно должна быть задана макрокоманда SE#QU.

LE#SA — читать очередную базовую запись. С помощью этой макрокоманды можно организовать последовательную или частично-последовательную обработку базовых массивов. Обычно для задания начальной точки обработки используются макрокоманды BE#SL или BE#AD. Если эти макрокоманды не были использованы, LE#SA при первом к ней обращении считывает запись с минимальным ключом. Дополнительно нужно употреблять макрокоманду SE#QU.

ZU#SA — поместить запись в базовый массив. Макрокоманда позволяет расположить на прежнее место запись, считанную одной из предыдущих команд. Необходимая информация берется из префикса. Дополнительно используется макрокоманда SE#QU.

SE#QU — определить подпрограммы обработки базовых массивов. В операндах макрокоманды указывается количество обрабатываемых базовых массивов (один или два) и виды операций. Допустимыми кодами операций являются ZUSA, BESL, BEAD. Макрокоманда SE#QU не должна встречаться в последовательности выполнимых команд, так как она состоит из набора подпрограмм для выполнения функций чтения и записи для базовых массивов и получает управление от четырех предыдущих макро-

команд. В операндах рекомендуется задавать только те коды операций, которые действительно используются в данной программе, так как это влияет на размеры программы. Подпрограмма последовательного чтения включается всегда, в связи с чем в операндах макрокоманды SE#QU эта функция не указывается. Как правило, макрокоманда SE#QU должна записываться в конце программы и перед ней должен ставиться оператор Ассемблера LTORG, а непосредственно после SE#QU записываться макрокоманда DA#NA для каждого базового массива.

DA#NA — передать управление по концу массива. В операндах задаются имя массива (ровно семь знаков) и символический адрес перехода в случае достижения конца. Макрокоманда должна записываться непосредственно после SE#QU.

DE#RO — открыть массивы; DA#BS — закрыть массивы. Эти две макрокоманды позволяют открывать и закрывать не только массивы банка данных, но и другие массивы, организованные средствами ДОС. В операндах макрокоманды перечисляются вначале массивы ДОС (не более пяти), а затем массивы банка данных. Для массивов ДОС в макрокоманде задаются имена описаний массивов (имена DTF), для массивов банка данных — имена префиксов. Наличие круглых скобок в макрокомандах обязательно.

KE#TT — получить цепь записей. Макрокоманда позволяет получить записи массивов связей, относящихся к одному предмету или рабочему месту. Одновременно с записями связей можно получать и подчиненные базовые записи. Операнды макрокоманды записываются в формате ключевых слов. Операндом ANKADR задается поле в системной части базовой записи, содержащее начальный адрес цепи. Базовая запись, содержащая начальный адрес цепи, должна к моменту входа в макрокоманду KE#TT располагаться в оперативной памяти. Операнд FOLGADR задает поле в записи связи, содержащее адрес следующей записи связи (адрес продолжения цепи). В операнде ARBERV указывается имя префикса для массива связей. Если кроме записи связи требуется прочитать подчиненную базовую запись, нужно задать операнды BASADR и ARBERB. Первый операнд указывает поле в записи связи, содержащее адрес подчиненной базовой записи, второй — имя префикса для подчиненного базового массива. Операнд ROUT определяет имя подпрограммы обработки записей. Обращение к подпрограмме происходит после считывания каждой записи связи (или пары записей, если заданы BASADR и ARBERB).

Возврат из подпрограммы должен выполняться по регистру 14 (BR14). При достижении конца цепи вместо обращения к подпрограмме управление передается следующей за макрокомандой KE#TT команде. В подпрограмме необходимо предусмотреть сохранение содержимого регистра 14, в противном случае продолжение обработки цепи становится невозможным.

RU#PI — выполнить операцию ввода-вывода. Макрокоманда предназначена для выполнения операций записи, чтения записей,

преобразования адресов. Операция задается первым операндом, который может принимать значения: DERO — открыть массив; DABS — закрыть массив; BDIA — читать базовую запись по адресу; BSLA — читать базовую запись по ключу; BFSA — писать базовую запись; VDIA — читать запись связи по адресу; VFSA — писать запись связи; VERD — уплотнить адрес; ERWE — расширить адрес. Для чтения записей в префиксе предварительно необходимо поместить ключ или адрес. Для помещения адреса из четырехбайтового поля в запись в девятибайтовое поле в префиксе надо указать вторым операндом имя префикса, а третьим — имя четырехбайтового поля в записи. Первый операнд должен иметь значение ERWE. Для обратной пересылки адреса (из префикса в запись) используется операция VERD. Для других операций третий операнд не задается.

При считывании запись помещается в область, расположенную вслед за префиксом, для обратного помещения записи в массив запись берется из этой же области.

ST#AR — назначить базовые регистры. Макрокоманда назначает базовые регистры, загружает их, резервирует область сохранения. Первый операнд указывает от одного до шести базовых регистров, второй операнд задает начальный адрес программы, третий указывает на необходимость резервирования области сохранения (да или нет). Если третий операнд не задан, область сохранения резервируется по умолчанию.

TE#XT — установить связь с оператором. Макрокоманда позволяет выдавать из программы сообщения или указания оператору на пишущую машинку и принимать ответ. Текст сообщения задается первым операндом. Это может быть символический адрес начала текста в программе или непосредственно сам текст, заключенный в кавычки. Второй операнд указывает адрес поля, в которое помещается ответ оператора. Операнд задается только в случае, если требуется ответ. Третьим операндом можно указать длину выдаваемого текста, если она не определяется характеристикой длины имени или текст содержит внутри кавычки.

AD#RU — вывести на печать. Макрокоманда предназначена для вывода на устройство широкой печати произвольной информации. Операнды макрокоманды записываются в формате ключевых слов и могут принимать следующие значения:

DEF=JA или NEIN. При DEF=JA в макрокоманде определяются массив печати, области ввода и вывода и производится открытие массива печати. При DEF=NEIN определение и открытие массива необходимо предусмотреть в своей программе;

DBER=AT#DBER или символическое имя. Если операнд опущен, области печати присваивается имя AT#DBER. Перед использованием макрокоманды в область печати необходимо заслать подлежащую выводу информацию (строку), при этом необходимо учитывать, что первый байт области печати содержит символ управления печатью;

DRULNG=p. Этот операнд задает длину строки. Если операнд опущен, длина строки принимается равной 121 символу (первый символ является символом управления печатью);

EINHEIT=SYSnnn. Операнд задает символическое имя устройства печати. Если операнд опущен, принимается SYSLST;

ZEIL=p. Операнд указывает число печатаемых строк на один лист. Если операнд опущен, число строк принимается равным 62;

UEB=(имя 1, имя 2, ...). Этим операндом задаются имена строк, печатаемых в верхней части каждого листа (в шапке листа). Длины строк принимаются равными характеристикам длины перечисленных в операнде символических имен. Длина первой строки шапки не должна превышать 105 символов, если задан операнд DATUM. Если операнд UEB опущен, шапка листа не печатается;

LEERZ=(n1, n2, ...). Операнд задает число пропускаемых строк после каждой строки шапки, а количество интервалов не должно быть больше трех. Число подоперандов должно соответствовать числу строк шапки. Если операнд не задается, между строками шапки делается один интервал;

DATUM=JA или NEIN. При DATUM=JA в первой строке шапки каждого листа печатается текущая дата, начиная со 106 позиции. При DATUM=NEIN дата не печатается. Стандартное значение — JA.

SEITE=JA или NEIN. При SEITE=JA в следующей податой строке шапки печатается номер листа. Стандартное значение — JA.

REG=11 или г. Этим операндом указывается регистр возврата, если AD#RU используется как подпрограмма. При опущенном операнде принимается REG=11, если макрокоманда встречается в программе более одного раза, в противном случае после выполнения AD#RU управление получает следующая за ней команда. Перед первым обращением к AD#RU необходимо в первый байт области печати поместить символ «1» (с помощью команды MVI AT#DBER, C'1') для подвода бумаги к началу листа, а в поле AT#SEI поместить упакованный ноль (по команде ZAP AT#SEI,=P'0'), чтобы нумерация листов начиналась с единицы.

Пример построения программы с использованием макрокоманд системы БАСТАИ дается ниже.

			Номер оператора
GETI	ST#AR	(3,4),X'1000',JA	1
	DE#RO	(MK),(MP,TM,PM)	2
	GET	MK,R	3
	MVC	GD#SCHL,R	} 4
	BE#SL	MP	
	MVI	AT#DBER,C'1'	} 5
	ZAP	AT#SEI,P'0'	
	MVC	DETAL,GD#GIDNR	
	KE#TT	ANKADR=GD#ADEAS,	* 6

			Номер оператора
		FOLGADR=AD#ADNAS,	*
		BASADR=AD#ADUBS	*
		ARBERB=RM,	*
		ARBERV=TM,	*
		ROUT=PP1	
		GET1	
END	B	MK	7
	CLOSE	DABS,MP	8
	RU#PI	(TM,RM)	9
	DA#BS	'КОНЕЦ РАБОТЫ'	10
	TE#XT		
	EOJ		
PP1	MVC	AT#DBER+1(120),=120X'40'	
	MVC	AT#DBER+10(5),MD#MIDNR	
	MVC	AT#DBER+20(3),AD#ABGNR	
	ST	14,R14	
	BAL	7,ADRU	
	L	14,R14	
	BR	14	
ADRU	AD#RU	DEF=JA,	*
	*	UEB=(ST1,DETAL),	*
		LEERZ=(1,2),	*
		REG=7	
MP	VA#RB	GD#,PREDMET,10	11
	DS	CLn СИСТЕМНАЯ ЧАСТЬ	12
	DS	CLm ЧАСТЬ ПОЛЬЗОВАТЕЛЯ	13
RM	VA#RB	MD#,RABMEST,5	
	.		
	.		
TM	VA#RB	AD#,MARSCHR	
	.		
	.		
МК	DTFCD	EOFADR=END,	*
	
ST1	DS	0CL50	
	DC	10C''	
	DC	C'МАРШРУТ ОБРАБОТКИ ДЕТАЛИ'	
	DC	16C''	
DETAL	DS	CL10	
R14	DS	F	
R	DS	80C	
	LTORG		
	SE#QU	2,BESL	14
	DA#NA	PREDMET,END	15
	DA#NA	RABMEST,END	16
	END		

В примере показаны последовательность и варианты использования макрокоманд. По оператору 1 оформляется начало программы, т. е. устанавливается начальный адрес программы и производится загрузка базовых регистров. Оператор 2 предназначен для открытия входного массива на перфокартах и трех массивов банка данных. По оператору 3 производится считывание кода предмета с перфокарты, а с помощью операторов 4 читается по ключу запись из массива предметов. Оператор 5 подготавливает информацию к выводу на печать. Оператор 6 позволяет после-

довательно получать пары записей, относящихся к найденному с помощью операторов 4 коду предмета из массива технологических маршрутов и массива рабочих мест. Найденная пара записей обрабатывается подпрограммой PPI. В ней формируется строка для вывода на печать, производится обращение к подпрограмме печати, заданной с помощью макрокоманды AD#RU, и осуществляется возвращение в макрокоманду KE#TT (по регистру 14). Оператор 7 выполняет закрытие массива на перфокартах, а операторы 8 и 9 — закрытие массивов банка данных. Вместо операторов 7—9 можно было использовать макрокоманду DA#BS и в ней перечислять имена всех массивов, как это сделано в операторе 2. Операторы 11, 12 и 13 описывают запись массива предметов. В конкретном случае вместо операторов 12 и 13 необходимо определить в отдельности каждое поле в системной части и каждый реквизит в части пользователя. Аналогичным образом описывается структура двух остальных массивов банка данных. В операторе 14 задается число обрабатываемых в программе базовых массивов, а параметром BESL указывается, что в программе используется макрокоманда BE#SL. Операторы 15 и 16 задают адрес перехода в случае, если в процессе работы программы обнаружен конец массива.

В подпрограмме печати код детали печатается во второй строке шапки, а в строках таблицы, кроме данных из записей рабочих операций, содержатся и коды рабочих мест. Это становится возможным благодаря заданию в макрокоманде KE#TT операндов BASADR и ARBERB, в результате чего при каждом входе в подпрограмму PPI в областях RM и TM содержится пара соответствующих записей. Следует особо отметить, что для выполнения макрокоманд системы БАСТАЙ при редактировании программы в нее необходимо включить модули ввода-вывода, использованные для загрузки банка данных: EAWURZEL, EAARBEIT, EAEROABS. Макрокоманды ST#AR, AD#RU и TE#XT вспомогательные и никакой связи с модулями ввода-вывода не имеют, поэтому их можно использовать в любых программах.

§ 7. Программирование специальных программ пользователя (СПП)

Специальные программы пользователя (СПП) предназначены для дополнительной обработки записей массивов при загрузке, изменениях и реорганизации базы данных. СПП пишутся на языке Ассемблера. При генерации модулей СПП вставляются в нужные места программы, и поэтому они оформляются в виде ее части. При обращении к этим программам управление из модулей передается первой команде СПП. В конце работы специальная программа пользователя должна передать управление команде, следующей за последней командой или константой СПП, например:

		Номер оператора	
SPPNN	CLC	FK#ANZ,C1	1
	BNE	ENDE	2
	MVC	KENNZ (5),C2	3
	B	ENDE	4
C1	DC	C'LAD'	5
C2	DC	C'ЗАГР.'	6
	DS	ØH	7
ENDE	EQU	*	8

Выполнение приведенной в примере СПП может быть закончено по оператору 2 или 4. И в том и в другом случае управление должна получить команда, следующая за последним оператором СПП (в примере это реализовано с помощью оператора EQU). Оператор 7 нужен для того, чтобы последующие команды были выравнены по границе полуслова. Для сокращения СПП рекомендуется использовать в ней константы и поля записывать в специальной области между параметрами DFKONSTF и DEFINEND. В приведенном примере операторы 5 и 6 можно записать в специальной области для констант, а операторы 4 и 7 в этом случае не нужны.

При программировании СПП необходимо помнить, что каждая СПП должна начинаться параметром ANSAROnn и заканчиваться параметром ENSAROnn. Вместо текста СПП между этими двумя параметрами рекомендуется записать оператор Ассемблера COPY, а текст СПП включить в виде книги в библиотеку исходных модулей, например:

```
ANSAROnn      COPY SPP3
ENSAROnn
```

Если при программировании СПП допущены ошибки, то таким способом можно обойтись без повторной генерации модуля, достаточно лишь внести изменения в текст СПП в библиотеке исходных модулей и заново оттранслировать сгенерированную ранее программу.

При программировании СПП для модуля LEITPROG следует соблюдать следующие ограничения на применение регистров:

регистры 2, 3, 8 и 10 используются в модуле LEITPROG как базовые;

регистры 9, 12 (в случае их использования) нужно запоминать при входе в СПП и восстанавливать при выходе;

регистр 7 используется в СПП10 для специальных целей (см. ниже);

регистры 0, 1, 13, 14 и 15 используются макрокомандами DOS.

На применение остальных регистров ограничения не накладываются. В СПП для модулей реорганизации массивов можно применять регистры с 5-го по 9-й включительно.

Рассмотрим назначения СПП и особенности их программирования.

В модуле LEITPROG используются СПП с номерами 02, 04, 05, 06, 15, 17, 18 и 19, предназначенные для распечатки записей при загрузке и изменениях массивов банка данных:

СПП02 — печать записей при загрузке МП;

СПП04 — печать записей при изменении МП;

СПП05 — печать строк заголовка при загрузке и изменении МС;

СПП06 — печать записей при загрузке и изменении МС;

СПП15 — печать записей при загрузке РМ;

СПП17 — печать записей при изменении РМ;

СПП18 — печать шапки при загрузке и изменении ТМ;

СПП19 — печать записей при загрузке и изменении ТМ.

Предполагается, что печать записей выполняется с помощью макрокоманды AD#RU (подробно макрокоманда AD#RU будет описана далее).

В СПП05 и СПП18 необходимо предусмотреть формирование строк для вывода оглавления документа и прогона бумаги к началу листа. Каждая спецификация или последовательность записей рабочих операций, относящихся к одному и тому же предмету, должна распечатываться с нового листа. Это можно выполнить с помощью следующих операторов:

		Номер оператора
ZAP	AT#SEI,=P'G'	1
MVI	BER,C'1'	2
MVC	S1,...	3
MVC	S2,...	4
.		
MVC	SN+K(L),GD#GIDNR	5

Оператор 1 устанавливает номер первого листа спецификации или последовательности операций в единицу. Оператор 2 осуществляет прогон бумаги к началу листа (область печати имеет имя BER). Операторы 3, 4 ... формируют строки оглавления документа (в макрокоманде AD#RU задан операнд UEB=(S1, S2, ...)). Оператор 5 помещает код предмета длиной L байт в строку SN, начиная с позиции K. Если в оглавлении документа нужно поместить другие данные из записи предмета, например наименование, эти данные можно взять из соответствующего поля области, определенной параметром DFSATZGD. Следует заметить, что здесь только формируются данные для печати оглавления. Непосредственно печать происходит в СПП06 или в СПП19 при первом к ним обращении или при переходе на новый лист. В этих программах пользователя необходимо сформировать строку для печати в области, заданной параметром DBER в макрокоманде AD#RU, и обратиться к подпрограмме печати. При формировании строки следует учитывать, что первый символ области печати является управляющим символом и его здесь изменять

нельзя. Необходимые данные для формирования строки находят-ся в областях, описанных параметрами DFSATZSD и DFSATZGD для СПП06 или DFSATZAD и DFSATZMD для СПП19. Подпро-грамму печати удобнее всего помещать в специальной области для констант и полей (см. параметр DFKONSTF). Подпрограмму печати можно написать в виде:

DRUCK	AD#RU	DEF=YA, DBER=BER,	*
		EINHEIT=SYS005,	*
		UEB=(S1, S2, ...),	*
		LEERZ=(1, 1, ... 2),	*
		REG=5	
BER	DS	0CL121	
	DS	CL1	
BER1	DS	CLn	
	.		
	.		
S1	DS	CL120	
S2	DS	CL120	
	.		
	.		

Строку для печати перед обращением к подпрограмме необхо-димо помещать начиная с адреса BER1; команда обращения к подпрограмме печати имеет вид BAL 5, DRUCK. Параметр REG должен быть задан обязательно.

В СПП02 и СПП15, как и в двух предыдущих СПП, также необходимо сформировать строку и обратиться к подпрограмме печати. Данные для формирования строки берутся из области, определенной параметром DFSATZGD для МП или DFSATZMD для РМ. Дополнительно в этих СПП нужно предусмотреть при первом входе в них команды подготовки, аналогичные командам, описанным для СПП05, с номерами 1—4.

СПП04 и СПП17 предназначены для печати записей при изме-нении массива предметов и массива рабочих мест. Для формиро-вания и печати строк в этих СПП рекомендуется строить обра-щения к СПП02 и СПП15 соответственно. В последних необходи-мо предусмотреть возврат в вызвавшую их СПП. С учетом выше-изложенного СПП02 или СПП15 могут выглядеть примерно так:

ANSARO02			Номер оператора
SPP02	LA	6,AUSG	1
EING4	BC	0,NECHST	2
	ZAP	AT#SEI,=P'0'	
	MVI	BER,C'1'	
	MVC	S1,...	
	.		
	.		
	MVI	EING4+1,X'F0'	3

			Номер оператора
NECHST	MVC	BER1,...	
	.		
	.		
	BR	6	4
AUSG	EQU	*	
ENSARO02			
ANSARO04			
SPP4	BAL	6,EING4	5
ENSARO04			

При первом обращении к СПП02 оператор 2 пропускается и выполняются команды подготовки. Оператор 3 предназначен для изменения оператора 2 с целью обхода команд подготовки при последующих обращениях к СПП02. Оператор 4 при обращении к СПП02 из модуля LEITPROG передает управление следующей за ним команде, так как оператор 1 загружает в регистр 6 адрес следующей за оператором 4 команды.

При обращении к СПП02 из СПП04 управление получает оператор 2, а в регистр 6 загружается адрес следующей за оператором 5 команды, поэтому управление из СПП02 передается не в LEITPROG, как в первом случае, а в СПП04, а оттуда уже в LEITPROG.

Во всех СПП для печати записей можно выбирать из поля FK#ANZ сокращенное название выполняемой функции длиной в три байта и печатать его в строке.

СПП03 подготавливает записи для загрузки или добавляет их в массив предметов. Входная запись находится в области, определенной параметром DFEBERGD. В этой программе необходимо перенести данные из входной записи в область, определенную параметром DFSATZGD. Код предмета нужно поместить в системную часть записи (в поле GD#GIDNR). Если выходная запись состоит из нескольких входных, обращение к данной СПП производится каждый раз после считывания входной записи. При этом определяется, какая запись прочитана и в какие поля выходной записи нужно поместить данные из только что прочитанной входной записи.

СПП16 komponует записи при загрузке или добавлении в РМ. Входная запись определена параметром DFEBERMD, выходная — параметром DFSATZMD. СПП16 составляется аналогично СПП03.

СПП07 подготавливает записи при загрузке или добавлении в МС. Данные пересылаются из входной записи DFEBERSD в выходную DFSATZSD. Дополнительно пересылается контрольная часть поля, заданного параметром SDIDNRGP, в поле SD#PROBS и часть поля, заданного параметром SDIDNRGK, — в поле SD#PRUBS. Если выводная запись состоит из нескольких входных, нужно поступать как в СПП03.

СПП20 аналогична СПП07, только используется она при загрузке или добавлении в ТМ. Входная запись определена пара-

метром DFEBERAD, выходная — параметром DFSATZAD. Часть поля, определенного параметром ADIDNRGD, нужно переслать в поле AD#PROBS, а номер рабочей операции — в поле AD#ABGNR. Часть поля, заданного параметром ADIDNRMD, следует поместить в поле AD#PRUBS (если ТМ связан с РМ).

Для выполнения корректировки массивов из модуля LEITPROG происходит обращение к следующим СПП: СПП08 — для МП, СПП09 — для МС, СПП21 — для РМ, СПП22 — для ТМ.

В момент обращения к СПП корректирующая (входная) запись находится в области, определенной параметрами DFEBERGD, DFEBERSD, DFEBERMD или DFEBERAD, подлежащая корректировке запись — в соответствующей рабочей области, определенной параметрами DFSATZGD, DFSATZSD, DFSATZMD или DFSATZAD. В СПП необходимо перенести корректирующие данные из входной записи в соответствующую рабочую область. Во входной записи поля, которые не должны корректироваться, рекомендуется заполнять пробелами, а в СПП делать соответствующую проверку: если поле во входной записи заполнено пробелами, оно не должно пересылаться в рабочую область.

СПП10 вызывается непосредственно после считывания входной записи для ее обработки: можно проверить запись на полноту информации, установить или изменить признак функции, для записей связей проверить наличие соответствующих записей в базовых массивах и т. д. Если в записи обнаружена ошибка, можно выдать соответствующее сообщение и исключить ошибочную запись, для чего необходимо передать управление по адресу, содержащемуся в регистре 7, после чего будет прочитана следующая входная запись.

СПП11 организует запись контрольных точек. Если на сообщение об ошибке ВАА035D дается ответ «4», происходит обращение к СПП11, где с помощью макрокоманды СНКРТ можно выдать запись контрольной точки, а затем распечатать содержимое области памяти с помощью макрокоманды PDUMP. После выхода из СПП11 все массивы закрываются. При повторном запуске с контрольной точки выполнение программы продолжается с адреса, следующего за PDUMP.

СПП12 и СПП23 предназначены для распаковки кода предмета или рабочего места при выводе сообщений об ошибках на пишущую машинку, если этот код хранится в записи в упакованном виде (см. параметры CDPROTOК и MDPROTOК в приложении 2). В регистре 1 находится адрес поля, содержащего упакованный код. Распакованный код нужно поместить в поле TD#PROGD для МП или TD#PROMD для РМ, длина этого поля задается параметром GDPROTOК или MDPROTOК соответственно. Эти СПП нужно обязательно программировать, если код в записи находится в упакованном виде.

СПП13 и СПП24 позволяют заменять признаки сортировки и код компоненты (для МС) или рабочего места (для ТМ). Входная запись для замены должна содержать старые и новые значе-

ния заменяемых полей. Новые значения полей из входной записи, находящейся в области, заданной параметром DFEBERSD или DFEBERAD, необходимо поместить на место старых значений, находящихся в рабочей области, определенной параметрами DFSATZSD или DFSATZAD. Если заменяется код компоненты или рабочего места, нужно часть кода (контрольную часть) поместить в поле SD#PRUBS для MC или в поле AD#PRUBS для TM.

В СПП25 программируется закрытие дополнительных массивов, если предыдущие СПП использовали дополнительные массивы. Обращение к СПП25 происходит в тот момент, когда для всех массивов достигнуто состояние «конец массива».

Большинство из описанных для модуля LEITPROG СПП не обязательно и их можно не программировать. Параметры ANSAROnn и ENSAROnn для отсутствующей СПП в этом случае опускаются. Обязательно программируются СПП12 и СПП23, если код предмета и код рабочего места содержатся в записях на диске в упакованном виде. Непременно должны также программироваться СПП для выполнения загрузки или изменения, если в параметрах для LEITPROG были указаны эти функции.

Для модуля BDAREORG используются СПП31—СПП34.

СПП31 предназначена для открытия дополнительных массивов, используемых при реорганизации. Обращение к ней происходит непосредственно после открытия участвующих в реорганизации массивов банка данных. Здесь нужно открыть все массивы, описанные между параметрами DFANWDAT и DEFINEND.

СПП32 изменяет базовый массив в процессе реорганизации. Обращение к этой программе происходит после считывания каждой записи исходного массива. В СПП нужно перенести запись из области ввода в область вывода, при этом допускается изменить структуру записи, исключить отдельные поля (реквизиты) или добавить новые. В область вывода следует переслать все поля, в том числе код предмета или код рабочего места и начальные адреса цепей. Для стирания записи константу C'SS' помещают в поле, заданное параметром RAKENNZF в выходной записи, а добавляемые записи — в область вывода, при этом в поля для начальных адресов цепей помещают константу C'ENDE'. Добавляемые записи берутся из дополнительного массива, момент добавления должен определяться в СПП путем сравнения ключевых реквизитов (кода предмета или рабочего места) из входной записи и записи дополнительного массива. Входная запись описывается параметром DFEINBAD, выходная — параметром DFAUSBAD. Если параметры опущены (это разрешается), то первый байт входной записи имеет имя BR#ESABR, выходной — имя BR#ASABR.

СПП33 дает возможность распечатывать записи выходного (реорганизованного) массива непосредственно перед их загрузкой. Массив печати определяет параметр DFANWDAT, открытие и закрытие массива печати производится в СПП31 и СПП34. Под-

лежащая печати запись находится в области вывода. Область печати (строка) задается параметром DFKONSTR. В СППЗЗ можно организовать вывод реорганизованного массива не только на печать, но и на любой другой носитель, например на ПК или МЛ (это необходимо в случае, если опущен параметр BDALADEN):

В СППЗ4 нужно предусмотреть закрытие дополнительных массивов, которые были открыты в СППЗ1. Обращение к СПП происходит после закрытия участвующих в реорганизации массивов банка данных.

Для модуля VDAREORG применяются СПП41—СПП43.

СПП41 аналогична СППЗ1, но она используется только при реорганизации массива связей.

В СПП42 программируется закрытие дополнительных массивов, открытых в СПП41.

СПП43 дает возможность изменять массивы связей во время их реорганизации. Обращение к ней происходит после считывания каждой входной записи связей. При этом доступна также и старшая базовая запись (запись МП). Входная запись связи находится в области, определенной параметром DFEINVED, выходную запись нужно поместить в область, определенную параметром DFAUSVED, старшая базовая запись находится в области, определенной параметром DFUEBBAD (если эти параметры опущены, записи располагаются в областях, имеющих имена VR#ESABR, VR#ASABR и VR#BSABR соответственно).

В СПП43 можно предусмотреть выполнение таких функций, как изменение структуры записей, удаление отдельных полей (реквизитов), добавление новых полей и новых записей. При добавлении новых записей группы записей связей, относящиеся к одной старшей базовой записи, вызываются в порядке возрастания кодов базовых записей, а внутри группы — в порядке возрастания признаков сортировки, поэтому записи в дополнительном массиве должны быть рассортированы в таком же порядке. Если для МС задан параметр DISPSTUF, то спецификации (группы записей) вызываются в порядке возрастания ступеней диспозиции сборочных единиц, внутри каждой ступени — в порядке возрастания кодов сборочных единиц. Так как наряду с записью связи всегда доступна и старшая базовая запись, в этой СПП можно запрограммировать также распечатку записей при реорганизации базы данных.

ГЛАВА 2

ОРГАНИЗАЦИЯ И ВЕДЕНИЕ НЕЗАВИСИМЫХ МАССИВОВ ИНФОРМАЦИИ

§ 1. Назначение системы программирования САВИ

Анализ информационной базы АСУП показал, что отдельные массивы данных логически не взаимосвязаны между собой и их нецелесообразно обрабатывать с помощью специализированного банка данных. Такие данные наиболее удобно организовывать в виде независимых массивов, организация и ведение которых сводится к стандартным процедурам обработки информации: ввод, контроль, компоновка, сортировка, корректировка и распечатка. Эти процедуры могут быть выполнены универсальными или специализированными системами программирования. Одной из них является САВИ, которая была использована авторами при создании АСУП на Первом московском часовом заводе.

САВИ представляет собой независимую от решаемых проблем систему программ организации и ведения массивов, построенную по модульному принципу. В состав системы входят четыре группы модулей: управляющие модули, интерпретаторы инструкций языка процедур, рабочие модули для выполнения функций САВИ, вспомогательные модули.

Управляющие модули берут на себя функцию вызова и управления работой исполнительных модулей. Модули-интерпретаторы анализируют инструкции языка процедур, проводят синтаксические и смысловые проверки и создают управляющие таблицы. Рабочие модули реализуют функции определения, накопления и поиска данных.

Вспомогательные модули предназначены для ввода и вывода данных, корректировки ошибок, протоколирования и т. д. Все модули располагаются в библиотеке абсолютных модулей на резидентном пакете дисков в виде отдельных программных фаз и занимают 15 цилиндров.

Система САВИ ориентирована на работу под управлением

ДОС ЕС и может использоваться на всех ЕС ЭВМ со следующей минимальной конфигурацией: центральный процессор с МОЗУ в 64 К байт; консоль (устройство связи с оператором); системный магнитный диск (МД); МД для массива таблиц данных (ТОД), рабочего массива, основного массива; устройство для ввода управляющих карт; устройство параллельной печати (АЦПУ).

Для управления процессами создания и обслуживания массивов система использует специальный язык описания данных и процедур. Элементами языка являются инструкции, ключевые операции и зарезервированные слова (см. приложение 5).

Из отдельных элементов языка составляют предложения, описывающие характеристики элементов данных (реквизитов), состав и структуру массивов, а также операции обработки данных. Описания массивов преобразуются в таблицы описания данных, которые записываются в массив ТОД на магнитном диске.

При создании, изменении и обработке массивов нет необходимости заново описывать их структуру, так как все требуемые для выполнения этих функций данные берутся из ТОД. Имена необходимых ТОД указываются в соответствующих инструкциях.

В системе САВИ используются следующие типы массивов: входной, основной, выходной, массив ТОД и рабочий.

Входной, основной и выходной массивы являются информационными массивами пользователя, т. е. их структуру устанавливает сам пользователь с учетом требований ДОС ЕС, рабочий массив и массив ТОД системные.

Рабочий массив должен обязательно находиться на МД и занимать минимум 300 дорожек. Он предназначен для хранения промежуточных результатов при выполнении функций определения и поиска.

В массиве ТОД содержатся все ТОД, составленные пользователем. Можно создать несколько массивов ТОД, причем в один массив ТОД может быть включено максимум 190 ТОД. Минимальный объем памяти на МД для массива ТОД — два цилиндра. Так как массив ТОД является системным массивом, а пользователю необходимо часто получать сведения о содержании ТОД, система САВИ обеспечивает доступ к массивам ТОД с помощью вспомогательных функций (см. приложение 6).

Входные массивы могут подготавливаться на перфокартах, перфолентах, магнитных лентах или дисках. Входные массивы на перфоносителях называются первичными. В первичном входном массиве допускается иметь до 150 реквизитов, разделенных при необходимости на 20 сегментов (не больше) с различными типами записей. Для массивов на магнитных носителях (вторичные массивы) допускаются записи только одного типа.

Логическую структуру записи пользователь устанавливает в соответствующей ТОД, физическая длина записи ограничивается на перфокарте 80 байтами, на МД — 3600 байтами.

Основной массив — это массив, созданный с помощью системы программирования САВИ на магнитных носителях. На МД дан-

ный массив может иметь последовательную или индексно-последовательную организацию данных, на МЛ — только последовательную с метками или без меток. В основном массиве должны быть записи одного типа. В конце записей массивов с последовательной организацией допускается текстовая часть переменной длины.

Выходные массивы создаются при поиске в основном массиве и содержат результаты поиска. За один прогон программы можно получить до 255 выходных массивов на различных носителях данных. Результаты поиска накапливаются в рабочем массиве, поэтому число и величина выходных массивов зависит от величины рабочего массива. Выходные массивы создаются всегда как массивы с последовательной организацией данных.

При работе системы САВИ массивам назначаются следующие логические устройства: SYS004 — для массива ТОД; SYS005 — для рабочего массива; SYS006 — для основного массива; SYS007 — для выходного массива; SYS008 — для входного массива на МЛ или ПЛ; SYS009 — для сменного устройства при вводе с перфокар.

При выполнении функции обработки для выходных массивов можно назначать несколько логических устройств с номерами SYS007 — SYSmax.

В управляющих картах //DLBL в качестве имени файла необходимо задавать SWDBT для массива ТОД и SWARB — для рабочего массива. Имена файлов в картах //DLBL и //TLBL для входных, основных и выходных массивов состоят из шестисимвольного имени ТОД и следующего за ним символа 0 (нуль) или 1 (один). Нуль добавляется к имени ТОД для входных и основных массивов, единица — для выходных массивов.

Для массивов на магнитных лентах с метками задается карта //LBLTYP TAPE, для массивов с индексно-последовательной организацией данных на магнитных дисках //LBLTYP NSD (03).

Последовательность создания и обслуживания массивов с помощью системы САВИ представлена на рис. 8.

§ 2. Создание таблиц описания данных

Для каждого создаваемого с помощью системы САВИ информационного массива необходимо подготовить две таблицы описания данных: одна предназначена для описания входного массива, вторая — основного. Для выходных массивов отдельная ТОД не требуется, так как все необходимые данные содержатся в ТОД основного массива.

Таблицы описания данных составляются с помощью инструкций языка описания данных (см. приложение 6). ТОД входных массивов составляются на основании схем (макетов) перфорации первичных документов. К схемам перфорации системой САВИ предъявляются следующие требования:

ведущие реквизиты (ключи сортировки) в записях должны размещаться либо в начале записи, либо в конце, либо в начале и

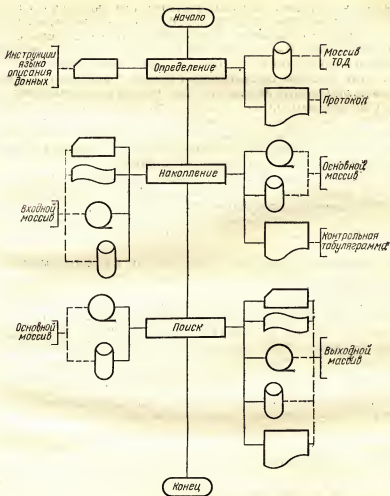


Рис. 8. Последовательность создания и обслуживания локальных массивов с помощью системы САВИ

в конце, но не должны появляться между другими (неведущими) реквизитами;

в случае, если предусматривается изменение массива с помощью инструкции AENDERN, в записях необходимо предусмотреть наличие реквизита «признак функции изменения», который должен находиться в области ключей.

В ТОД входного массива должно быть описано не менее двух областей: область ключей и информационная область, причем в последней описывается как минимум один сегмент. В ТОД описываются также с помощью соответствующих инструкций все

реквизиты входного массива. Ниже приводится пример написания ТОД для входного массива.

```
___DEFINIEREN__A01101___;  
___LADEN___;  
___DATE1__TYP=PRIMAER___;  
___BEREICH__ANFANG=1__LAENGE=20___;  
___FELD__NAME=M1__LAENGE=3___;  
___FELD__NAME=F__LAENGE=1___;  
___FELD__NAME=CECH__LAENGE=6___;  
___FELD__NAME=UCHASTOK__LAENGE=10___;  
___BEREICH__ANFANG=21__LAENGE=14___;  
___SEGMENT___;  
___FELD__NAME=OBOZN__LAENGE=10___;  
___FELD__NAME=KOLICH__LAENGE=4__DARST=N___;  
___IDENT__SATZ=CECH,UCHASTOK  
___FUNKTION=F___;
```

Здесь описана ТОД входного массива A01101, находящегося на ПК и подготовленного для записи в массив ТОД. Массив состоит из двух областей: первая область начинается с первой колодки ПК и описывает макет (M1), поле функции (F) и ведущие реквизиты (CECH, UCHASTOK); вторая область начинается с колонки 21, ее длина 14 байт. Область описывает сегмент с количественными реквизитами (OBOZN, KOLICH). Инструкция IDENT задает имена полей, по которым рассортировывается массив. FUNKTION задает имя поля функции.

В ТОД для основного массива инструкции описания областей и сегментов не задаются. Здесь приводится структура массива на МЛ или МД и инструкции, описывающие виды контроля и подготовки реквизитов (полей) для вывода. Пример ТОД для основного массива выглядит следующим образом:

```
___DEFINIEREN__BAAA51___;  
___LADEN___;  
___DATE1__BLOCKFAKTOR=20___;  
___FELD__NAME=S__LAENGE=1___;  
___FELD__NAME=CECH__LAENGE=6___;  
___TESTEN__MASKE=999999___;  
___FELD__NAME=UCHASTOK__LAENGE=10___;  
___TESTEN__MASKE=999999999___;  
___FELD__NAME=OBOZN__LAENGE=10___;  
___FELD__NAME=KOLICH__LAENGE=3__DARST=P___;  
___AUFBER__MASKE='___?@??'___;  
___IDENT__SATZ=CECH,UCHASTOK  
___ZUSTAND=S___;
```

Основной массив получается из входного массива, описанного в предыдущем примере. Длина блока равна 20 записям. Содержание поля CECH будет проверяться на цифровое содержание при

загрузке информации входного массива. Инструкция AUFBER указывает, что при выводе значение реквизита KOLICH будет отредактировано по шаблону 99,99; ZUSTAND описывает имя поля состояния, согласно содержанию которого производится коррекция массива.

Подготовленные таким образом таблицы описания данных переносятся на перфокарты и помещаются в массив ТОД на диске. Управляющие карты для включения ТОД в массив приводятся ниже.

```
// JOB DEFIN
// ASSGN SYS005,X'nnn'
// DLBL SWARB,,0,DA } рабочий массив
// EXTENT SYS003,...
// ASSGN SYS004,X'nnn'
// DLBL SWDBT,'имя массива ТОД',,DA } массив ТОД
// EXTENT SYS004,...
// UPSI nn
// EXEC SW#SAWI
<таблица описания данных>
/*
/ &
```

Нулевой и первый биты байта UPSI управляют выводом протокола и построением массива ТОД. Если нулевой бит равен нулю, в протокол включаются все инструкции из ТОД, если он равен единице, в протокол выдаются только инструкции, содержащие ошибки.

Первый бит байта UPSI должен быть равен единице, если массив ТОД создается (ранее не существовал), и нулю, если ТОД включается в уже существующий массив ТОД.

§ 3. Создание и ведение массивов

В системе САВИ создание и ведение массивов производится с помощью таблиц описания данных, которые формируются для каждого обрабатываемого массива. Это означает, что массив ТОД должен быть создан заблаговременно.

С помощью инструкций создания и ведения массивов (см. приложение 7) пользователь может построить на МЛ или МД основной массив и постоянно его обновлять. При загрузке этого массива выполняются: ввод входного массива; проверка входного массива на формальную правильность; подготовка входных данных; вывод проверенных и подготовленных данных в основной массив.

Система контролирует порядок рассортированности входного массива при его создании и ведении. Контроль порядка сортировки записей можно исключить с помощью карты UPSI. Во время загрузки массивов могут выполняться подпрограммы контроля и преобразования данных, если это предусмотрено ТОД.

Для обслуживания основного массива система выполняет:

расширение (ERWEITERN) — в созданный массив вводятся новые записи (входные записи записываются в заданной последовательности в конец массива);

замену (ERSETZEN) — заменяются отдельные поля или записи;

добавление (ZUFUEGEN) — в основной массив добавляется одна или несколько записей;

стирание (LOESCHEN) — записи удаляются из основного массива и исключаются из дальнейшей обработки;

консервацию (STILLEGEN) — временно приостанавливается использование записей (законсервированные записи при необходимости могут использоваться при дальнейшей обработке);

активизацию (AKTIVIEREN) — процедура, обратная консервации, т.е. представление законсервированных записей для дальнейшей обработки;

корректировку (BERICHTIGEN) — внутри одной основной записи изменяется один или несколько реквизитов (исходный рекуизит при этом заменяется новым).

Система обслуживает только те массивы, записи которых рассортированы в порядке возрастания идентификационных признаков. Исключение из этого правила допускается при выполнении функции расширения.

Система предусматривает два способа обслуживания массивов: за один прогон программы выполняется одна из функций; за один прогон программы выполняется несколько функций, их вызов производится с помощью инструкции AENDERN; нужная функция определяется на основании содержания поля функции во входной записи.

В процессе загрузки введенные данные обрабатываются подпрограммами контроля и преобразования, заданными в ТОД к основному массиву. Входные записи с ошибками фиксируются в протоколе и в построении основного массива не участвуют. Если входным является массив, построенный системой САВИ, т.е. вторичный, то при построении основного массива учитывается поле его состояния. Логически стертые записи при загрузке пропускаются.

Для выполнения функций накопления пользователь системы должен задать последовательность выполнения описанных инструкций. При этом необходимо соблюдать следующие правила: каждая последовательность инструкций для выполнения накопления должна начинаться инструкцией SPEICHERN, которая сообщает имя используемой ТОД и при необходимости коды пользователя;

если предусмотрено включение дополнительных модулей, после управляющей инструкции накопления следуют соответствующие инструкции EINFUEGEN (максимум две на каждую точку выхода), в которых называются представляемые в распоряжение модуля поля и данные пользователя;

следующая инструкция задает одну из функций накопления, после которой могут использоваться инструкции ZUORDNEN.

Ниже приводятся примеры заданий для выполнения функций загрузки и корректировки массивов.

```
// JOB LADEN загрузка
// ASSIGN SYS005,X'nnn'
// DLBL SWARB,0,DA } рабочий массив
// EXTENT SYS005,...
// ASSIGN SYS004,X'nnn'
// DLBL SWDBT,'<имя массива ТОД>',DA } массив ТОД
// EXTENT SYS004,...
// ASSIGN SYS006,X'nnn'
// DLBL BAAA511,'AAA51' } основной массив
// EXTENT SYS006,...
// UPSI nn
// EXEC SW#SAWI
SPEICHERN BAAA51 AAAA AUS A01101 BBBB;
LADEN;

/*
<исходные данные на ПК>
/*
/&

// JOB AENDERN изменение
// ASSIGN SYS005,X'nnn'
// DLBL SWARB,0,DA } рабочий массив
// EXTENT SYS005,...
// ASSIGN SYS004,X'nnn'
// DLBL SWDBT,'<имя массива ТОД>',DA } массив ТОД
// EXTENT SYS004,...
// ASSIGN SYS006,X'nnn'
// DLBL BAAA510,'AAA51' } основной массив (старый)
// EXTENT SYS006,...
// ASSIGN SYS007,X'nnn'
// DLBL BAAA511,'AAA51' } основной массив (скорректированный)
// EXTENT SYS007,...
// EXEC SW#SAWI
SPEICHERN BAAA51 AAAA AUS A01101 BBBB;
AENDERN;

/*
<ПК корректуры>
/*
/&
```

Карты корректуры выполнения этого задания необходимо заранее рассортировать в порядке возрастания ключевых реквизитов. Если скорректированный и старый основной массив находятся на одном и том же пакете дисков, им должны быть назначены разные участки и в картах //DLBL указаны различные идентификаторы.

Если основной массив организуется индексно-последовательным методом доступа, в задание необходимо вставить карту //LBLTYP NSD и соответствующим образом изменить карты //DLBL и //EXTENT. Если основной массив должен иметь последовательную организацию и записывается на магнитную ленту с метками, в задание нужно вставить карты //TLBL и //LBLTYP TAPE.

Управление выводом протокола и проверкой рассортированности входных записей осуществляется с помощью карты URSl. Если нулевой бит равен нулю, в протокол печатаются все входные записи, если он равен единице — только ошибочные записи. Если первый бит байта UPSI равен нулю, рассортированность записей входного массива не проверяется, если равен единице, проверка производится, и следующие не в порядке рассортированности записи печатаются в протокол ошибок и не включаются в основной массив.

§ 4. Использование массивов

Инструкции поиска системы САВИ позволяют выполнять простейшие операции обработки данных, содержащихся в основных массивах, такие, как подсчет выходных записей, удовлетворяющих заданным условиям поиска, сортировка массивов, вывод массивов на печать или на другие машинные носители данных, суммирование реквизитов, преобразование реквизитов при выводе. Подробное описание инструкций поиска приведено в приложении 9.

Ниже приводятся примеры заданий для сортировки и распечатки основного массива.

```
// JOB SORT сортировка
// ASSGN SYS001,X'280'
// ASSGN SYS002,X'280'
// MTC REW,SYS002
// ASSGN SYS003,X'191'
// DLBL SORTWK1,,0
// EXTENT SYS003,...
// ASSGN SYS004,X'nnn'
// DLBL SWDBT,'<имя таблицы ТОД>','DA' } массив ТОД
// EXTENT SYS004,...
// ASSGN SYS005,X'nnn' }
// DLBL SWARB,,0,DA } рабочий массив
// EXTENT SYS005,...
// ASSGN SYS006,X'nnn' }
// DLBL BAAA510,'AAA51' } старый массив
// EXTENT SYS006,...
// ASSGN SYS007,X'nnn' }
// DLBL BAAA511,'AAA51' } рассортированный массив
// EXTENT SYS007,...
// EXEC SW#SAWI
// SUCHEN BAAA51;
// BAAA511 AUSGEBEN ALLE AUF PLATTE;
// AUFBEREITEN BLOCK 20;
// SORT CEHC UCHASTOK;

/*
/*

// JOB DRUCKEN распечатка массива
// ASSGN SYS004,X'nnn'
// DLBL SWDBT,'<имя таблицы ТОД>','DA' } массив ТОД
// EXTENT SYS004,...
// ASSGN SYS005,X'nnn' }
// DLBL SWARB,,0,DA } рабочий массив
// EXTENT SYS005,...
```

```

// ASSGN SYS006,X'nnn'
// DLBL BAAA510,'AAA51' } основной массив
// EXTENT SYS006,...
// EXEC SW#SAWI
  SUCHEN BAAA51 ;
  AUSGEBEN ALLE ;
  ZAEHLEN ;
  KOPF 'форма 0101101' 2'массив 0101151' 30 ;
  VORSCHUB 1 ;
  KOPF 'Ведомость оборудования' 10 ;
  VORSCHUB 1 ;
  KOPF '-----' 1 ;
  KOPF 'код :код :обозначение : ' 1 ;
  ' количество станков : ' 35 ;
  VORSCHUB 1 ;
  KOPF ' цеха :участка : '
  ' : ' 35 ;

  VORSCHUB 1 ;
  KOPF '-----' 1 ;
  VORSCHUB 1 ;
  KOPF ;
  ADDIEREN KOLICH IN KOL ;
  ZEILE CECH 3 UCHASTOK 11 OBOZN 20 KOLICH 37 ;

/*
/&

```

При выводе массива на печать необходимо следить за тем, чтобы длина строки на устройстве печати не превышала 120 символов.

§ 5. Подготовка исходных данных

Исходные данные в системе САВИ могут быть подготовлены на перфокартах и перфолентах. При использовании первых каждая строка исходного документа переносится на одну или несколько (но не более пяти) перфокарт. В случае, если запись занимает несколько перфокарт, нужно предусмотреть, чтобы на каждой перфокарте в одной и той же колонке содержался последовательный номер перфокарты в записи. Ведущие (ключевые) реквизиты и признак функции изменения должны повторяться на каждой ПК.

Применение перфоленты позволяет вести подготовку данных в формате с разделителями. Разделители полей (реквизитов), сегментов и записей выбираются либо стандартные (запятая, точка с запятой и точка соответственно), либо из ТОД входного массива (см. приложение 6). Информация на перфоленте может быть подготовлена для ввода в любом коде, преобразование ее во внутренний код ЭВМ осуществляется в процессе ввода данных по специально подготовленным таблицам. Таблицы должны быть оформлены в виде фазы в библиотеке абсолютных модулей и иметь имя SW#TLBEM, если выбранный код ПЛ содержит символ переключения, и имя SW#TLBEO — в противном случае.

При подготовке информации для загрузки (первоначального создания) массивов в поле признака функции необходимо перфорировать нуль. Если данные предназначены для изменения, признак функций выбирается согласно приведенному в описании инст-

рукции AENDERN. В записях, предназначенных для загрузки или добавления, нужно перфорировать все реквизиты, в записях на стирание (удаление), консервацию и активизацию достаточно указать только признак функции и ключевые реквизиты, в записях на корректировку (замену) — признак функции, ключевые и изменяемые реквизиты. Если выполнение функции AENDERN для некоторого массива не предусматривается, то признак функции во входных записях не требуется.

Системой САВИ предусматривается также использование в качестве первичных носителей данных магнитных лент и дисков. В этом случае входными массивами системы САВИ могут служить массивы, созданные другими системами программирования. Для таких массивов с помощью инструкций языка описания данных подготавливаются ТОД входного и основного массивов.

ГЛАВА 3

ОСОБЕННОСТИ РЕШЕНИЯ ЗАДАЧ В УСЛОВИЯХ СУЩЕСТВОВАНИЯ БАНКА ДАННЫХ

В этой главе описывается опыт разработки, внедрения и функционирования АСУП, использующей организацию фонда нормативно-справочной информации (НСИ) в виде банка данных, на примере решения следующих задач:

- расчет применяемости деталей и сборочных единиц в изделии;
- расчет специфицированных и сводных норм расхода материалов;

- расчет сводных трудовых нормативов;

- расчет нормативных калькуляций.

Отличительными особенностями выбранных задач, на основании которых можно судить о характере применения банков данных в АСУП, являются:

- использование большей части исходной информации;

- сильная зависимость времени решения этих задач от структуры и методов построения информационной базы;

- применение результатов решения данных задач в большинстве задач АСУП.

Ознакомимся с фондом НСИ АСУ Первого московского часового завода АСУ-1-МЧЗ, рассмотрим состав и структуру информационной базы данных и ее связь с вышеуказанными задачами, приведем описание алгоритмов их решения. При этом не ставится цель подробно изложить все особенности выполняемых расчетов, перечислить все процедуры, так как имеется достаточно литературы, где подробно рассматриваются организационно-экономическая сущность и алгоритмы расчетов большинства типичных задач АСУП, в том числе и нижерассматриваемых [9]. Мы ограничимся описанием лишь основных расчетов по каждому из выходных документов, что позволит достаточно полно представить проблемы, встающие перед разработчиками программного обеспечения, уяснить пути их решения и степень использования (заимствования) соответствующего программного обеспечения. Хотелось бы под-

черкнуть, что основная цель нашей работы — обратить внимание разработчиков и пользователей АСУП на те принципиальные особенности проектирования в условиях существования банка данных, которые существенным образом влияют на методологию проектирования АСУП на базе ЭВМ третьего поколения.

§ 1. Описание фонда нормативно-справочной информации АСУ-1-МЧЗ

Фонд НСИ АСУ-1-МЧЗ включает информацию 40 первичных нормативно-справочных документов (НСД), заполняемых и используемых различными службами предприятия. Остановимся кратко лишь на тех из них, информация которых вошла в банк данных, так как они представляют наибольший интерес с точки зрения организации информационной базы АСУ. Перечень их приведен в табл. 4, а сами формы приводятся по тексту (кроме единичных и групповых спецификаций, которые приняты согласно ГОСТу 2. 108—68, ГОСТ 2. 113—70).

Таблица 4

Документ	Условный индекс формы	Служба, выпускающая документ
Маршрутная карта	Г	Отдел главного технолога (ОГТ)
Комплектовочная карта	Б	То же
Состав и характеристика оборудования	К	»
Справочник групп оборудования	Т	ВЦ
Станкозатраты на деталь	Л	ОГТ
Ведомость подетальных норм расхода материала	Д	»
Ценник на отходы	И	ВЦ
Справочник наименований изделий и их составных частей	П	Отдел главного конструктора (ОГК)
Номенклатура-ценник материалов	Ж	ВЦ
Справочник групп материалов по сводной номенклатуре	Р	»
Справочник наименований инвентарных объектов основных фондов	М	»
Справочник потерь на полуфабрикаты	С	»
Спецификация (единичная)	А	ОГК
Групповая спецификация	В	»

Маршрутная карта (форма № 01) содержит описание технологического процесса и межцеховых переходов изготовления детали (сборочной единицы) по всем операциям в технологической последовательности с указанием соответствующих данных по оборудованию, трудовым и другим нормативам.

Графы «Код детали (сборочной единицы)», «Код варианта исполнения», «Код единицы нормирования», «Код цеха», «Код участка», «Номер операции», «Наименование операции», «Обо-

Наименование детали (сборочной единицы)			Код детали (сборочной единицы)	Код варианта исполнения	Код единицы нормирования	МАРШРУТНАЯ КАРТА								
Код цеха	Код участка	Номер операции	Наименование операции	Обозначение технологического документа	Оборудование		Код			Норма обслуживания станков	Разряд работ	Норма выработки, шт./ч	Норма времени, мин	Расценка, руб.
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

начение технологического документа», «Наименование оборудования» и «Код оборудования» заполняются в отделе главного технолога на основании операционных карт ранее разработанного технологического процесса, а остальные графы — в отделе труда и заработной платы на основании цеховых ведомостей подетальных норм расценок.

Маршрутная карта формируется по всей номенклатуре деталей (сборочных единиц), изготавливаемых заводом, а также на детали (полуфабрикаты), получаемые по кооперированным поставкам, и на покупные изделия, если они подвергаются доработке. Она содержит полный перечень операций обработки детали (сборочной единицы), начиная с заготовительной операции по всему межцеховому технологическому маршруту.

Признак окончания технологического маршрута обозначается номером операции 990, который проставляется вслед за последней технологической операцией маршрута, завершающей обработку детали (сборочной единицы).

Для деталей (сборочных единиц), не имеющих вариантов исполнения, в графе «Код варианта исполнения» проставляется 00.

Для деталей (сборочных единиц), имеющих вариант исполнения, маршрутная карта составляется следующим образом. Если для всех вариантов исполнения нет различия в технологическом процессе, то составляется маршрутная карта базового варианта с обозначением варианта исполнения 00. Если в зависимости от варианта исполнения имеются различия в технологическом процессе, то на первые операции обработки, не зависящие от варианта исполнения, составляется маршрутная карта базового варианта с обозначением варианта исполнения 00 без указания признака окончания, а на все последующие операции — отдельные маршрутные карты на каждый вариант. Эти маршрутные карты будут иметь признак окончания обработки (номер операции 990).

В графе «Код единицы нормирования» проставляется код, указывающий, на какое количество установлены нормы времени и

расценки. Если детали не отличаются конструктивно друг от друга, но изготавливаются по разным технологиям, то маршрутная карта составляется на один, основной технологический процесс, наиболее целесообразный с технологической и экономической точки зрения. На параллельные технологические процессы изготовления детали (сборочной единицы), не содержащие конструкторских отличий, в маршрутной карте, в правом верхнем углу проставляется «ПВ» (параллельный вариант).

Код цеха и участка указывается по классификатору структурных подразделений завода. Номер операции состоит из трех знаков: первые два составляют порядковый номер операции, а третий — заполняется нулями.

Третий (нулевой) разряд предусмотрен для обозначения новых операций, возникающих при изменении технологического процесса.

Графа «Код оборудования» для данной операции определяется технологом. Код оборудования проставляется на основании заводского кодификатора. При выполнении операции вручную в графе «Код оборудования» маршрутной карты проставляется код ручной работы.

Признак передачи готовой детали или сборочной единицы на сборку или на испытания обозначаются после строки окончания обработки номером «999». В графе «Код цеха» по этой строке записывается код цеха или код склада, куда поступает деталь, сборочная единица.

По деталям (сборочным единицам), приобретаемым со стороны и проходящим дальнейшую технологическую обработку в цехах завода, первым цехом-заготовителем записывается служба Отдела внешней кооперации под номером операции «000».

Технолог, заполнив свою часть маршрутной карты, передает ее в ОТиЗ. ОТиЗ заполняет свою часть в строгом соответствии с технологическим процессом.

Реквизиты «Код тарифной сетки», «Код вида норм» заполняются согласно заводским кодификаторам.

Норма многостаночного обслуживания в маршрутной карте указывается в физических единицах оборудования. Если рабочий по норме обслуживает один станок, то в графу «Норма обслуживания станков» проставляется 1, если два станка — 2 и т. д.

Реквизит «Разряд работ» проставляется в соответствии с действующей на заводе тарифной сеткой.

В графе «Норма времени, мин» записывается норма времени платежная до сотых долей минуты. В графу «Расценка, руб.» записывается платежная расценка до десятой доли копейки. Для удобства внесения изменений в маршрутную карту между наименованием операций целесообразно оставлять одну пустую строку.

Заполненный документ возвращается в ОГТ, который рассылает копии маршрутной карты во все службы завода, где она используется. В ВЦ данные этого документа перфорируются согласно схеме перфорации, приведенной в табл. 5. При перфорации рекви-

зит «Код варианта исполнения» объединен с реквизитом «Код деталн (сборочной единицы)» и на схемах перфорации здесь и далее отдельно не показывается. Структура объединенного реквизита приведена на с. 72—73.

Таблица 3

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код деталн (сборочной единицы)	4—16	9(13)
Код единицы нормирования	17	9(1)
Код цеха	18—19	9(2)
Код участка	20—21	9(2)
Номер операции	22—24	9(3)
Код оборудования	25—29	9(5)
Код тарифной сетки	30	9(1)
Код вида норм	31	9(1)
Норма обслуживания станков	32—33	9(2)
Разряд работы	34	9(1)
Норма времени в минутах	35—42	9(6), 9(2)
Расценка	43—49	9(4), 9(3)

Форма № 02

Наименование изделия (сборочной единицы)		Код изделия (сборочной единицы)	Код варианта исполнения	КОМПЛЕКТОВОЧНАЯ КАРТА		
Код цеха	Номер операции	Детали (сборочные единицы), входящие в комплект				
		наименование	код	код варианта исполнения	количество	код цеха поставщика
1	2	3	4	5	6	7

Комплектовочная карта (форма № 02) предназначена для записи данных о деталях, сборочных единицах и покупных деталях, входящих в комплект собираемой единицы.

Комплектовочная карта для изделий и сборочных единиц с вариантом исполнения составляется для базового варианта с кодом варианта исполнения «00». Такая карта распространяется на все спецификации вариантов и получает при этом все варианты из спецификации.

В комплектовочной карте операции сборочных работ должны быть проставлены по всем операциям технологического процесса.

Графа «Код цеха» заполняется на основании классификатора структурных подразделений.

Номер операции проставляется в технологической последовательности их выполнения согласно данным маршрутной карты.

Графы «Код детали (сборочной единицы), входящей в комплект» и «Количество» заполняются в полном соответствии с конструкторской спецификацией.

Для деталей собственного производства в графе «Код цеха-поставщика» записывается или код склада, или код цеха. Если же деталь покупная, то ставится код отдела внешней кооперации (ОВК). Информация перфорируется согласно схеме перфорации, приведенной в табл. 6.

Таблица 6

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код детали (сборочной единицы)	4—16	9(13)
Код цеха	17—18	9(2)
Номер операции	19—21	9(3)
Код детали (сборочной единицы)	22—34	9(13)
Количество	35—36	9(2)
Код цеха-поставщика	37—38	9(2)

Форма № 03

Код цеха		СОСТАВ И ХАРАКТЕРИСТИКА ОБОРУДОВАНИЯ						
Код участка	Поточная линия		Оборудование участка (поточной линии)			Потери времени на переналадку, %	Потери времени на ремонт, %	
	наименование	код	модель	код	инвентарный номер		при двухсменной работе	при трехсменной работе
1	2	3	4	5	6	7	8	9

«Состав и характеристика оборудования» (форма № 03) заполняются по каждой единице оборудования, закрепленной за определенным цехом, и содержит нормативные данные, необходимые для расчета загрузки и пропускной способности оборудования. Документ формируется по рассчитываемому цеху в разрезе производственных участков по основному технологическому оборудованию.

В случае отсутствия поточной линии графа «Код поточной линии» заполняется нулями.

Реквизиты-признаки формы устанавливаются в соответствии с заводскими кодификаторами.

Графа «Потери времени на переналадку, %» заполняется в соответствии с утвержденными нормативами или на основании опытно-статистических данных.

В графы «Потери времени на ремонт, % при двухсменной работе» и «Потери времени на ремонт, % при трехсменной работе» записывается процент потерь рабочего времени на ремонт оборудования.

Информация перфорируется согласно схеме перфорации, представленной в табл. 7.

Таблица 7

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код цеха	4—5	9(2)
Код участка	6—7	9(2)
Код поточной линии	8—9	9(2)
Код оборудования	10—14	9(5)
Инвентарный номер оборудования	15—18	9(4)
Потери времени на переналадку, %	19—21	9(2), 9(1)
Потери времени на ремонт при двухсменной работе, %	22—24	9(2), 9(1)
Потери времени на ремонт при трехсменной работе, %	25—27	9(2), 9(1)

Форма № 04

Код цеха	СПРАВОЧНИК ГРУПП ОБОРУДОВАНИЯ			
Код группы	Оборудование		Наименование группы	Инвентарный номер
	модель	код		
1	2	3	4	5

Справочник групп оборудования (форма № 04) создается по каждому цеху и содержит в себе информацию о перечне единиц оборудования с учетом объединения их в группы.

Графы «Код группы» и «Код оборудования» заполняются в соответствии с заводским кодификатором. Новым группам оборудования коды присваиваются в ОГТ и по установленной форме передаются в ИВЦ.

Информация перфорируется согласно схеме перфорации, приведенной в табл. 8.

«Станкозатраты на деталь» (форма № 05) составляется для оборудования, расчет загрузки и пропускной способности которого производится по инвентарным номерам. Документ формируется при необходимости расчета загрузки оборудования по каждой его единице. Аналогичные данные, имеющиеся в маршрутной карте, в

Таблица 8

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код цеха	4—5	9(2)
Код группы оборудования	6—10	9(5)
Код оборудования	11—15	9(5)
Наименование группы оборудования	16—39	X(24)

Форма № 05

Код цеха		СТАНКОЗАТРАТЫ НА ДЕТАЛЬ								
Код участка	Оборудование			Деталь (сборочная единица)			Код варианта исполнения	Номер операции	Код единицы нормирования	Станкозатраты, мин
	код	инвентарный номер	модель	номер	наименование	код				
1	2	3	4	5	6	7	8	9	10	11

расчете не участвуют. Документ может содержать несколько строк для одной детали с одинаковым номером операции, выполняемой на одинаковом оборудовании с различными инвентарными номерами и различными станкозатратами на деталь.

По каждому варианту исполнения детали (сборочной единицы) заполняется отдельная строка с заполнением всех показателей.

Графа «Номер операции» проставляется в технологической последовательности выполнения операций.

В графе «Код единицы нормирования» указывается, на какое количество деталей ведется нормирование.

Таблица 9

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код цеха	4—5	9(2)
Код участка	6—7	9(2)
Код оборудования	8—12	9(5)
Инвентарный номер оборудования	13—16	9(4)
Код детали (сборочной единицы)	17—29	9(13)
Номер операции	30—32	9(3)
Код единицы нормирования	33	9(i)
Станкозатраты, мин	34—41	9(6), 9(2)

В графе «Станкозатраты, мин» проставляются станкозатраты с учетом единицы нормирования.

Информация перфорируется согласно схеме перфорации, представленной в табл. 9.

Ведомость «Подетальные нормы расхода основных материалов» (форма № 06) содержит информацию о нормах расхода основных материалов на детали, изготавливаемые на предприятии.

Форма № 06

ПОДЕТАЛЬНЫЕ НОРМЫ РАСХОДА ОСНОВНЫХ МАТЕРИАЛОВ																
Деталь		Материал		Расчетные данные		Весовые данные и норма расхода										
наименование	код	код варианта исполнения	код цеха	наименование	код	размер	код	вес исходного материала	код единицы измерения	чистый вес	норма расхода по технологии	норма расхода с учетом отходов в заготовительном отделении	процент технических потерь	норма расхода с учетом ТНП	код единицы нормирования	коэффициент использования
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

В графах 2 и 3 записываются коды в полном соответствии со спецификацией.

В графе 4 указывается код цеха-потребителя материалов.

В графе 6 проставляется код исходного материала — материала, предназначенного для изготовления деталей согласно стандартам или техническим условиям. В графе 8 указывается код материала по технологии. Графы 6 и 8 заполняются согласно заводскому классификатору материалов.

В графе 10 проставляются единицы измерения материалов согласно номенклатуре ценника материалов.

В графе 11 записывается чистый вес, определяющий количество материала, содержащегося в готовой детали.

В графе 12 указывается расчетная норма расхода, определяющая максимальное допустимое количество материала, необходимого для изготовления единицы продукции при заданных производственных условиях, а в графе 13 — норма расхода с учетом отходов в заготовительном отделении.

В графе 15 записывается норма расхода с учетом технологически неизбежных потерь (ТНП). Их процент устанавливается в ОГТ. Норма расхода определяется с учетом отходов в заготовительном отделении и процента ТНП.

В графе 16 указывается код единицы нормирования, на которую устанавливаются все нормативные данные.

Об изменении кода материала по технологии ОГТ извещает ОМТС, склады завода и ВЦ. Код новому материалу присваивает-

ся в ВЦ и рассылается по установленной форме во все службы, где он используется.

Информация перфорируется согласно схеме, приведенной в табл. 10.

Таблица 10

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код детали (сборочной единицы)	4—16	9(13)
Код цеха	17—18	9(2)
Код исходного материала	19—26	9(8)
Код материала по технологии	27—34	9(8)
Код единицы измерения	35—36	9(2)
Чистый вес	37—44	9(4), 9(4)
Норма расхода по технологии	45—52	9(4), 9(4)
Норма расхода заготовительного отделения	53—60	9(4), 9(4)
Норма исходного материала	61—68	9(4), 9(4)
Код единицы нормирования	69	9(1)

Ценник на отходы (форма № 07) содержит информацию о ценах на отходы основных материалов, которые учитываются при решении задач в АСУП. Форма составляется на отходы черных и цветных металлов.

Форма № 07

ЦЕННИК НА ОТХОДЫ

Отходы		Единица измерения		Цена, руб.	Обоснование цены
наименование	код	наименование	код		
1	2	3	4	5	6

Код отходов проставляется в соответствии с номенклатурой-ценником материалов по первым пяти знакам, указывающим группу материала.

Информация перфорируется согласно схеме перфорации, представленной в табл. 11.

Таблица 11

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код отходов	4—8	9(5)
Код единицы измерения	9—10	9(2)
Цена, руб.	11—13	9(1), 9(2)

«Справочник наименований изделий и их составных частей» (форма № 08) заполняется на основании «Спецификации» (ГОСТ 2.108—68, ГОСТ 2.113—70).

Форма № 08

**СПРАВОЧНИК НАИМЕНОВАНИЙ ИЗДЕЛИЙ
И ИХ СОСТАВНЫХ ЧАСТЕЙ**

Код изделия (сборочной единицы, детали)	Код варианта исполнения	Краткое наименование изделия (сборочной единицы, детали)
1	2	3

В строках графы «Краткое наименование изделия (сборочной единицы, детали)» записываются коды изделий, механизмов и внешних оформлений по нормам НИИЧАСПРОМА и их краткое наименование.

Информация перфорируется согласно схеме перфорации, приведенной в табл. 12.

Т а б л и ц а 12

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код изделия (сборочной единицы, детали)	4—16	9(13)
Краткое наименование изделия (сборочной единицы, детали)	17—36	X(20)

«Номенклатура-ценник материалов» (форма № 09) содержит информацию о потребляемых на предприятии основных и вспомогательных материалах, покупных полуфабрикатах и комплектующих изделиях. Материалы располагаются в документе по классификационным группировкам в последовательности деления крупных группировок на мелкие и последовательности кодов материалов (номенклатурных номеров).

Форма № 09

НОМЕНКЛАТУРА-ЦЕННИК МАТЕРИАЛОВ								
Краткое наименование и характеристика мате- риала (ГОСТ, ТУ, марка, сорт, размер)	Код			Цена, руб.				Номер прейску- рента
	материала	балансового счета	единицы из- мерения	197	197	197	197	
				г.	г.	г.	г.	
1	2	3	4	5	6	7	8	9

Для отличия полуфабрикатов в «Номенклатуре-ценнике материалов» в первой позиции графы «Краткое наименование и характеристика материала» проставляется символ «*».

Информация перфорируется согласно схеме перфорации, представленной в табл. 13.

Таблица 13

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Наименование материала	4—51	X(48)
Код материала	52—59	9(8)
Код балансового счета	60—63	9(4)
Код единицы измерения	64—65	9(2)
Цена, руб.	66—71	9(3), 9(3)

Справочник групп материалов по сводной номенклатуре (форма № 10) содержит информацию о составе материалов, входящих в сводную номенклатуру ОМТС.

Форма № 10

СПРАВОЧНИК ГРУПП МАТЕРИАЛОВ ПО СВОДНОЙ НОМЕНКЛАТУРЕ

Номер соответствия	Группа материала		Код единицы измерения	Диапазон кодов материалов (сырья, комплектующих изделий) по заводскому классификатору	
	краткое наименование	код		от	до
				5	6
1	2	3	4	5	6

В графе «Номер соответствия» проставляется порядковый номер строки по сводной номенклатуре.

В графе «Краткое наименование группы материала» записывается наименование материала по сводной номенклатуре, сокращенное в пределах предусмотренной разрядности.

В графе «Код группы материала» проставляется сокращенный код группы материалов.

В графе «Диапазон кодов сырья, материалов и комплектующих изделий по заводскому классификатору» указывается, какие материалы из номенклатуры-ценника относятся к определенным группам по сводной номенклатуре.

Информация перфорируется согласно схеме перфорации, приведенной в табл. 14.

Справочник наименований инвентарных объектов основных фондов (форма № 11) содержит перечень наименований инвентарных объектов и соответствующие им коды.

Таблица 14

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Номер соответствия	4—7	9(4)
Краткое наименование группы материала	8—31	X(24)
Код группы материала	32—37	9(6)
Код единицы измерения	38—39	9(2)
Код материала, входящего в группу (нижняя граница)	40—47	9(8)
Код материала, входящего в группу (верхняя граница)	48—55	9(8)

Форма № 11

СПРАВОЧНИК НАИМЕНОВАНИЯ ИНВЕНТАРНЫХ ОБЪЕКТОВ ОСНОВНЫХ ФОНДОВ

Краткое наименование инвентарного объекта	Модель, тип станка	Код инвентарного объекта
1	2	3

В графе «Краткое наименование инвентарного объекта» записывается наименование инвентарного объекта, сокращенное в пределах предусмотренной разрядности.

В графе «Модель, тип станка» указывается модель или тип оборудования. Для остальных основных фондов графа не заполняется.

В графе «Код инвентарного объекта» проставляется код инвентарного объекта основных фондов.

Информация перфорируется согласно схеме перфорации, приведенной в табл. 15.

Таблица 15

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код инвентарного объекта	4—8	9(5)
Модель, тип станка	9—18	X(10)
Краткое наименование инвентарного объекта	19—49	X(31)

Справочник потерь на полуфабрикаты (форма № 12) составляется на полуфабрикаты, поставляемые предприятию НИИчаспром. Процент технологических потерь устанавливается на камни и пружины согласно данным Научно-исследовательского института часовой промышленности (НИИчаспром).

СПРАВОЧНИК ПОТЕРЬ НА ПОЛУФАБРИКАТЫ

Деталь (полуфабрикат)		Технологические потери, %
наименование	код	
1	2	3

Информация перфорируется согласно схеме перфорации, представленной в табл. 16.

Таблица 16

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код детали (полуфабриката)	4—16	9(13)
Технологические потери, %	17—21	9(2), 9(2)
Наименование детали (полуфабриката)	22—41	X(20)

Схемы перфораций единичных и групповых спецификаций приведены соответственно в табл. 17 и 18.

Таблица 17

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код изделий (сборочной единицы)	4—16	9(13)
Код детали (сборочной единицы)	17—29	9(13)
Количество	30—31	9(2)

Таблица 18

Реквизит (поле)	Номера колонок на ПК	Шаблон
Макет	1—2	9(2)
Функция	3	9(1)
Код изделия (сборочной единицы)	4—16	9(13)
Код детали (сборочной единицы)	17—29	9(13)
Количество	30—31	9(2)

§ 2. Состав и структура массивов банка данных АСУ-1-МЧЗ

Наиболее значительную (около 70%) по объему часть фонда НСИ АСУ-1-МЧЗ составляют данные, содержащиеся в единичных и групповых конструкторских спецификациях и маршрутно-техно-

логических картах. Информация этих документов организована в виде банка данных, состоящего из четырех логически взаимосвязанных массивов (рис. 9), организация и ведение которых осуществляется системой БАСТАЙ. В банк данных включена также информация из других первичных документов, логически связанных со спецификациями и маршрутными картами. Данные остальных документов фонда НСИ организованы в виде локальных массивов с помощью системы САВИ и в работе не рассматривается. Всего в банк данных включена информация 14 нормативно-справочных документов.

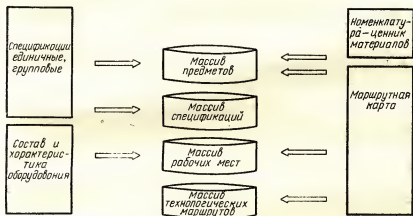


Рис. 9. Основные источники формирования банка данных на примере АСУ-1-МЧЗ

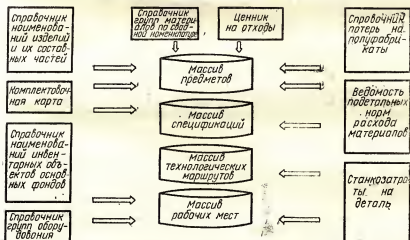


Рис. 10. Дополнительные источники формирования банка данных на примере АСУ-1-МЧЗ

Данные с дополнительных источников (рис. 10) вводятся в банк данных в режиме внесения изменений, информация основных источников служит для загрузки (первоначального создания) банка данных и дополнительно также может вводиться в режиме внесения изменений.

Как видно из рис. 9 и 10, в один массив банка данных может включаться информация из разных входных документов или же информация из одного входного документа поступать в разные массивы банка данных. Положение отдельных элементов данных и их связь в банке данных определяется в основном исходя из удобства обработки. Так, если рассматривать изделия состоящими не только из сборочных единиц и деталей, но и из материалов, из которых эти детали изготовлены, и установить в банке данных связи между информацией об изделиях, деталях и материалах, то это позволит совместить этап разузлования состава изделий с расчетом материальных нормативов. Непосредственный доступ к записям рабочих операций для каждой получаемой в процессе разузлования составной части изделия дает возможность рассчитывать заодно и трудовые нормативы.

Все массивы банка данных состоят из записей фиксированной длины. Для экономии места на диске записи объединены в блоки. Размеры записей и блоков приведены в табл. 19.

Таблица 19

Массив	Имя массива (имя DTF)	Размер записи, байт	Число записей в блоке
Массив предметов	PREDMET	125	13
Массив рабочих мест	RABMEST	60	23
Массив спецификаций	SPECIFI	33	34
Массив технологических маршрутов	MARKART	54	20

Записи массива банка данных, как указывалось выше, состоят из двух частей: системной части и области пользователя. В системной части содержатся данные, необходимые для накопления и поиска требуемой информации. Эти данные формируются программами системы БАСТАИ в момент загрузки массивов и корректируются при последующих изменениях. Для базовых массивов в системную часть включается также код предмета или рабочего места. В область пользователя вводятся все остальные данные, содержащиеся во входных документах. В пределах каждого массива системная часть записей имеет одинаковую структуру, в то время как область пользователя может быть различной структуры: она определяется соответствующим признаком или кодом предмета. Структура системной части массива предметов приведена в табл. 20, структура частей пользователя для записи деталей (сборочных единиц изделий) — в табл. 21, для записей полуфабри-

Таблица 20

Поле	Обозначение исходного документа	Шаблон	Тип поля	Длина поля, байт
Адрес следующей записи в массиве предметов	—	—	B	4
Код предмета	Ж, Г, А, В	9(13)	Z	13
Адрес первой записи спецификации	—	—	B	4
Счетчик записей в спецификации	—	—	B	2
Адрес первой записи цепи применения предметов	—	—	B	4
Счетчик записей в цепи применения	—	—	B	2
Степень диспозиции	—	9(2)	P	2
Адрес следующей записи предмета в рабочей (временной) цепи	—	—	B	4
Контрольное поле записи предмета в рабочей цепи	—	9(2)	Z	2
Текущий номер	—	—	B	2
Адрес записи первой рабочей опера- ции	—	—	B	4
Адрес записи последней рабочей опе- рации	—	—	B	4
Счетчик записей рабочих операций	—	—	B	2

Таблица 21

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Наименование детали, сборочной еди- ницы, изделия	П	A(20)	C	20
Признак детали (X'FF'), сборочной единицы (X'40')	—	—	B	1
Чистый вес	Д	9(4), 9(4)	P	5
Норма расхода материала по техно- логии	Д	9(4), 9(4)	P	5
Норма расхода материала для заго- товительного отделения	Д	9(4), 9(4)	P	5
Норма исходного материала	Д	9(4), 9(4)	P	5
Код единицы нормирования	Д	9(1)	Z	1
Код единицы измерения	Д	9(2)	Z	2
Код цеха	Д	9(2)	Z	2
Поле для хранения промежуточных результатов	—	—	—	30

катов — в табл. 22, для записей материалов и отходов — в табл. 23,
для записей групп материалов — в табл. 24.

Принятые в таблицах сокращения:

C — символичный (знаковый);

B — двоичный;

P — десятичный упакованный;

Z — зонный (десятичный распакованный).

Таблица 22

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Наименование полуфабриката	Ж	A(20)	C	20
Признак полуфабриката (X'FØ')	—	—	B	1
Пробелы	—	—	C	25
Процент технологических потерь	C	9(2), 9(1)	P	2
Код балансового счета	Ж	9(4)	Z	4
Код единицы измерения	И, Ж	9(2)	Z	2
Стоимость полуфабриката	И, Ж	9(3), 9(3)	P	4
Пробелы	—	—	C	18

Таблица 23

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Наименование материала	Ж	A(48)	C	48
Код балансового счета	Ж	9(4)	Z	4
Код единицы измерения	И, Ж	9(2)	Z	2
Цена материала	И, Ж	9(3), 9(3)	P	4
Код группы материалов	P	9(6)	Z	6
Пробелы	—	—	C	12

Таблица 24

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Номер соответствия	P	9(4)	Z	4
Наименование группы материалов	P	X(24)	C	24
Код единицы измерения	P	9(2)	Z	2
Код материала (нижняя граница)	P	9(8)	Z	8
Код материала (верхняя граница)	P	9(8)	Z	8
Пробелы	—	—	C	30

Таблица 25

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Адрес компоненты (что входит)	—	—	B	4
Контрольное поле компоненты	—	9(2)	Z	2
Адрес следующей записи спецификаций	—	—	B	4
Адрес записи сборочной единицы (куда входит)	—	—	B	4
Контрольное поле записи сборочной единицы	—	9(2)	Z	2
Адрес следующей записи в цепи применения предметов	—	—	B	4
Адрес предыдущей записи в цепи применения предмета	—	—	B	4

Во второй графе таблиц приведены сокращенные обозначения исходных документов (см. табл. 4), с которых переносятся значения реквизитов в определенное поле массива.

Структура системной части записей массива спецификаций приведена в табл. 25, части пользователя — в табл. 26.

Таблица 26

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Код цеха	Б	9(2)	Z	2
Номер рабочей операции	Б	9(3)	Z	3
Количество на сборочную единицу	А, Б, В	9(2)	P	2
Код цеха-поставщика	Б	9(2)	Z	2

Таблица 27

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Адрес следующей записи в массиве рабочих мест	—	—	В	4
Код рабочего места	К, М	9(9)	Z	9
Адрес первой рабочей операции в цепи использования рабочего места	—	—	В	4
Счетчик записей в цепи использования рабочего места	—	—	В	2

Таблица 28

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Код группы оборудования	Т	9(5)	Z	5
Наименование группы оборудования	Т	X(24)	C	24
Пробелы	—	—	C	12

Таблица 29

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Код цеха	К	9(2)	Z	2
Код участка	К	9(2)	Z	2
Код поточной линии	К	9(2)	Z	2
Процент потерь на переизладку	К	9(2), 9(1)	P	2
Потери на ремонт при двухсменной работе, мин	К	9(2), 9(1)	P	2
Потери на ремонт при трехсменной работе, мин	К	9(2), 9(1)	P	2
Пробелы	—	—	C	29

Структура системной части записей массива рабочих мест представлена в табл. 27, части пользователя для записей групп оборудования — в табл. 28, для записей характеристик оборудования — в табл. 29, для записей видов оборудования — в табл. 30.

Таблица 30

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Модель, тип оборудования	М	X(10)	C	10
Наименование оборудования	М	X(31)	C	31

Структура системной части записей массива технологических маршрутов приведена в табл. 31, части пользователя для записей маршрутных карт — в табл. 32, для записей станкозатрат — в табл. 33.

Таблица 31

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Адрес записи предмета	—	—	B	4
Контрольное поле записи предмета	—	9(2)	Z	2
Номер рабочей операции	Г, Л	9(7)	Z	7
Адрес записи следующей рабочей операции	—	—	B	4
Адрес записи предыдущей рабочей операции	—	—	B	4
Контрольное поле записи рабочего места	—	9(2)	Z	2
Адрес записи рабочего места	—	—	B	4
Адрес записи следующей рабочей операции в цепи использования рабочего места	—	—	B	4
Адрес записи предыдущей рабочей операции в цепи использования рабочего места	—	—	B	4

Таблица 32

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Код единицы нормирования	Г	9(1)	Z	1
Пробелы	Г	—	C	2
Код участка	Г	9(2)	Z	2
Код тарифной сетки	Г	9(1)	Z	1
Код вида норм	Г	9(1)	Z	1
Норма обслуживания	Г	9(2)	P	2
Разряд работы	Г	9(1)	Z	1
Норма времени, мин	Г	9(6), 9(2)	P	5
Расценка	Г	9(4), 9(3)	P	4

Таблица 33

Поле	Исходный документ	Шаблон	Тип поля	Длина поля, байт
Код единицы нормирования	Л	9(1)	Z	1
Код цеха	Л	9(2)	Z	2
Код участка	Л	9(2)	Z	2
Станкозатраты на деталь, мин	Л	9(6), 9(2)	P	5
Пробелы	—	—	C	9

Записи в базовых массивах МП и РМ рассортированы в порядке возрастания соответственно кода предмета и рабочего места. Записи внутри массива ТМ рассортированы по возрастанию номера рабочей операции, внутри массива МС — по возрастанию кода компоненты.

Между массивами рабочих мест и массивами технологических маршрутов существуют два вида связей: записи рабочих операций связаны адресными ссылками с записями групп оборудования, записи станкозатрат — с записями характеристик оборудования.

§ 3. Технология формирования базы данных (на примере АСУ-1-МЧЗ)

Информационная база АСУ-1-МЧЗ, как указывалось выше, организована в виде банка данных, состоящего из четырех массивов: массива предметов, массива рабочих мест, массива спецификаций, массива технологических маршрутов. Первые два массива базовые, два последних соединительные. Они содержат структурные связи между записями базовых массивов, поэтому соединительные массивы без базовых не могут быть построены и использованы.

Базовые массивы организованы по модифицированному индексно-последовательному методу и допускают обработку содержащейся в них информации методом последовательного и прямого доступа. Соединительные массивы созданы по методу прямого доступа и обработка содержащейся в них информации возможна только по адресам, находящимся в записях базовых массивов.

Таблица 34

Комплекс программ	Код формы НСД	Идентификатор массива НСИ
Создание массива предметов	07, 08, 09	PREDMET
Создание массива рабочих мест	03	RABMEST
Создание массива спецификаций	ГОСТ 2.108—68, ГОСТ 2.113—70	SPECIFI
Создание массива технологических маршрутов	01	MARKART
Изменение массива банка данных	01—12, ГОСТ 2.108—68, ГОСТ 2.113—70	PREDMET, SPECIFI, RAB- MEST, MARKART

Все программы для организации и ведения массивов банка данных объединены в комплексы, перечень которых приведен в табл. 34.

Программы комплексов создания массивов предназначены для первичного формирования массивов банка на базе исходных данных, подготовленных на перфокартах. Добавления и изменения в массивах выполняются программами комплекса «Изменение массивов банка данных».

Загрузка или изменение массивов банка данных производится с перфокарт. Каждая из них содержит макет перфорации и признак функции загрузки или варианта изменения. Возможные варианты функции и соответствующие им признаки приведены в табл. 35.

Таблица 35

Признак функции	Функция	Пояснения
0	Загрузка	Первоначальный ввод данных
1	Стирание	Удаление одной строки документа (одной записи массива)
2	Добавление	Введение дополнительно одной строки документа
3	Корректировка	Изменение значения реквизитов, не являющихся признаками
4	Групповое стирание	Удаление всех строк документов, относящихся к заданным реквизитам-признакам (например, удаление целой спецификации узла или всего технологического маршрута детали)
5	Замена	Замена адресных связей между записями базовых и соединительных массивов

Таблица 36

Код формы НСД	Макет перфорации	Функция					
		0	1	2	3	4	5
08	01	+	+	+	+	—	+
09	02	+	+	+	+	—	—
07	03	+	+	+	+	—	—
06	04	—	—	+	+	—	—
11	05	—	+	+	+	—	—
03	06	+	+	+	+	—	—
02	07	—	+	+	+	+	+
ГОСТ 2.108—68	08	+	+	+	+	+	+
ГОСТ 2.113—70	09	+	+	+	+	+	+
01	10	+	+	+	+	+	+
05	11	—	+	+	+	—	—
04	13	—	+	+	+	—	—
10	14	—	+	+	+	—	—
12	15	—	—	—	+	—	—

Соответствие форм документов, макетов их перфорации и допустимые для них функции отмечены знаком «+» в табл. 36.

Реквизиты входных форм делятся на обязательные (реквизиты-признаки) и необязательные (реквизиты-основания). При подго-

Таблица 37

Макет перфорации	Порядковый номер реквизита в схеме перфорации													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
01	+	+	+	—	—	—	—	—	—	—	—	—	—	—
02	+	+	—	+	—	—	—	—	—	—	—	—	—	—
03	+	+	+	—	—	—	—	—	—	—	—	—	—	—
06	+	+	—	—	—	+	+	—	—	—	—	—	—	—
08	+	+	+	+	+	—	—	—	—	—	—	—	—	—
09	+	+	+	+	+	—	—	—	—	—	—	—	—	—
10	+	+	+	—	+	+	+	+	—	—	—	—	—	—

Таблица 38

Макет перфорации	Порядковый номер реквизита в схемах перфорации													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
01	+	+	+	—	—	—	—	—	—	—	—	—	—	—
02	+	+	+	+	—	—	—	—	—	—	—	—	—	—
03	+	+	+	—	—	—	—	—	—	—	—	—	—	—
05	+	+	+	—	—	—	—	—	—	—	—	—	—	—
06	+	+	—	—	—	+	+	—	—	—	—	—	—	—
07	+	+	+	—	+	+	—	—	—	—	—	—	—	—
08	+	+	+	+	—	—	—	—	—	—	—	—	—	—
09	+	+	+	+	—	—	—	—	—	—	—	—	—	—
10	+	+	+	—	—	—	+	—	—	—	—	—	—	—
11	+	+	—	—	—	—	+	+	—	—	—	—	—	—
13	+	+	+	—	+	—	—	—	—	—	—	—	—	—
14	+	+	—	—	+	—	—	—	—	—	—	—	—	—

Таблица 39

Макет перфорации	Порядковый номер реквизита в схемах перфорации													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
01	+	+	+	—	—	—	—	—	—	—	—	—	—	—
02	+	+	—	+	—	—	—	—	—	—	—	—	—	—
03	+	+	+	—	—	—	—	—	—	—	—	—	—	—
04	+	+	+	—	—	—	—	—	—	—	—	—	—	—
05	+	+	+	—	—	—	—	—	—	—	—	—	—	—
06	+	+	—	—	—	+	+	—	—	—	—	—	—	—
07	+	+	+	—	+	+	—	—	—	—	—	—	—	—
08	+	+	+	+	—	—	—	—	—	—	—	—	—	—
09	+	+	+	+	—	—	—	—	—	—	—	—	—	—
10	+	+	+	—	—	—	+	—	—	—	—	—	—	—
11	+	+	—	—	+	+	+	+	—	—	—	—	—	—
13	+	+	+	—	+	—	—	—	—	—	—	—	—	—
14	+	+	—	—	+	—	—	—	—	—	—	—	—	—
15	+	+	+	—	—	—	—	—	—	—	—	—	—	—

товке исходных данных в зависимости от макета перфорации и признака функции некоторые реквизиты можно опускать (перфорировать на их месте пробелы). Обязательные реквизиты для функции загрузки (табл. 37), стирания (табл. 38), добавления и корректировки (табл. 39) и группового стирания (табл. 40) отмечены знаком «+», необязательные — знаком «—».

Таблица 40

Макет перфорации	Порядковый номер реквизита в схемах перфорации													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
07	+	+	+	—	—	—	—	—						
08	+	+	+	—	—	—	—	—						
09	+	+	+	—	—	—	—	—						
10	+	+	+	—	—	—	—	—	—	—	—	—	—	—

При выполнении загрузки и добавления опущенные при перфорации реквизиты заменяются в записях массивов банка данных пробелами. При корректировке опущенные реквизиты не изменяют прежнего содержимого полей в записях массивов банка данных; если же некоторый реквизит присутствует, прежнее содержимое поля заменяется новым.

Комплекс программ «Создание массива предметов» предназначен для загрузки в массив предметов информации форм № 08, 09 и 07. Перечень программ данного комплекса приведен в табл. 41.

Таблица 41

Программа	Имя программы	Объем памяти, байт
Контроль исходных данных	AA501	5 000
Сортировка	AA201	13 000
Подготовка данных для загрузки	AA502	15 000
Сортировка	AA202	13 000
Загрузка массива предметов	AA503	25 000

В блоке 1 (рис. 11) информация документов 08, 09 и 07 проверяется на правильность заполнения и перфорации. Обнаруженные в процессе контроля ошибки распечатываются на устройство SYSLST.

Полученный в блоке массив сортируется по возрастанию кода предмета (деталь, сборочная единица, материал, полуфабрикат, отход). Рассортированный массив в блоке 3 преобразуется к виду, удобному для загрузки, а именно: все записи приводятся к одинаковой длине, а коды предметов, в зависимости от входного документа (форма документа определяется по номеру макета перфорации), преобразуются к виду:

XXXXXXXXXXXX — для деталей и сборочных единиц;
 XXXXXXXXXX — для полуфабрикатов;

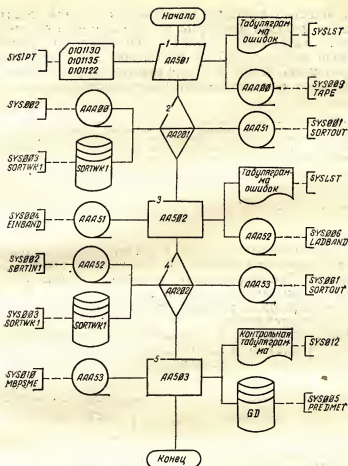


Рис. 11. Принципиальная блок-схема выполнения комплекса «Создание массива предметов GD (Gegenstanddatei)»

LLLLLXXXXXXXXX — для материалов (L — пробелы);
 LLLLLXXXXXLLL — для отходов.

Код предмета для деталей и сборочных единиц состоит из кода детали (сборочной единицы) и кода варианта исполнения (два последних знака кода).

Если при подготовке входных данных встречается запись детали (сборочной единицы) с кодом варианта исполнения, не равным нулю, то создается искусственная запись с нулевым вариантом исполнения, состоящая из кода предмета и символов XXXXX в поле «Название» (эта запись необходима в дальнейшем для правильной загрузки остальных массивов). Полученный массив сортируется по возрастанию реквизита «код предмета», и в блоке

5 производится запись массива на магнитный диск (загрузка). Все загружаемые записи распечатываются на устройстве SYS012. Так как для распечатки используется программа печати из комплекса «Изменение массива банка данных», на получаемой табуляграмме содержатся пустые поля, которые заполняются при загрузке или изменениях других массивов. Загрузка массивов предметов заканчивается выдачей на устройство SYSLOG сообщения BAA1001, на SYSLST при этом выдаются характеристики (параметры) загруженного массива.

Для загрузки 1000 записей, включая распечатку контрольной табуляграммы, требуется около 5 мин. Если загрузка прерывается (например, из-за сбоев машины), блок 5 выполняется повторно. Исходные данные для загрузки на магнитной ленте в процессе работы блока не изменяются.

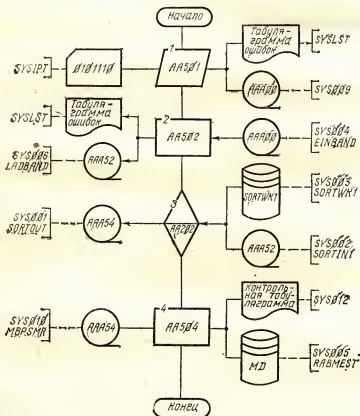


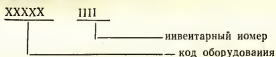
Рис. 12. Принципиальная блок-схема выполнения комплекса «Создание массива рабочих мест MD (Arbeitsplatzdatei)»

Комплекс программ «Создание массива рабочих мест» предназначен для первоначального ввода (загрузки) в массив рабочих мест информации формы № 01. Перечень программ комплекса дается в табл. 42, а блок-схема их выполнения представлена на рис. 12.

Таблица 42

Программа	Имя программы	Объем памяти, байт
Контроль исходных данных	AA501	5 000
Подготовка данных для загрузки	AA502	15 000
Сортировка	AA202	13 000
Загрузка массива рабочих мест	AA504	24 000

Процесс загрузки массива рабочих мест аналогичен процессу загрузки массива предметов. Код рабочего места состоит из двух реквизитов и имеет вид:



Все загружаемые записи выводятся в виде контрольной табуляграммы на устройство SYS012. Загрузка массива рабочих мест заканчивается выдачей на SYSLST характеристик загруженного массива, а на SYSLOG выдается сообщение BAA1001. Для загрузки 1000 записей, включая распечатку контрольной табуляграммы, требуется около 3 мин.

Комплекс программ «Создание массива спецификаций» предназначен для загрузки в массив спецификаций информации форм ГОСТ 2.108—68 и ГОСТ 2.113—70. Одновременно проводится проверка массива предметов на полноту и дополнение его недостающими записями. Перечень программ комплекса приведен в табл. 43.

Таблица 43

Программа	Имя программы	Объем памяти, байт
Контроль исходных данных	AA501	5 000
Подготовка данных для загрузки массива спецификаций и изменение массива предметов	AA505	35 000
Сортировка изменений	AA203	13 000
Уплотнение изменений	AA506	6 000
Изменение массивов банка	AA507	35 000
Сортировка	AA202	13 000
Загрузка массива спецификаций	AA508	20 000

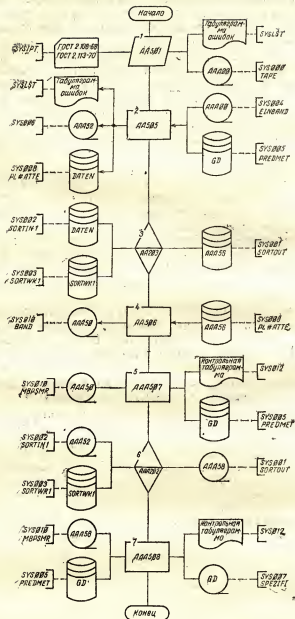


Рис. 13. Принципиальная блок-схема выполнения комплекса «Создание массива спецификаций SD (Stuecklistendatei)»

Массив спецификаций является соединительным: каждая запись этого массива отражает связь между парой записей массива предметов. По этой причине для создания массива спецификаций необходимо иметь массив предметов.

В блоке 1 (рис. 13) производится ввод исходных данных с перфокарт, контроль правильности перфорации и запись проверенных данных на магнитную ленту. В блоке 2 с магнитной ленты считывается очередная запись и проверяется наличие в массиве предметов записей для кода сборочной единицы (узла) и кода компоненты. Если в массиве предметов требуемая запись отсутствует, формируется искусственная запись для массива предметов, состоящая из кода предмета и символов XXXXX в поле «Наименование». Для деталей и сборочных единиц с ненулевым вариантом исполнения, что указывается двумя последними знаками кода, проверяется наличие искусственной записи, которая могла быть сформирована при загрузке массива предметов. В противном случае искусственная запись создается данным блоком, а в качестве ее признака в поле «Наименование» записываются символы ***. Искусственные записи выводятся на магнитный диск и в табуляграмму ошибок на устройство SYSLST, записи спецификаций — на магнитную ленту, используемую в дальнейшем для загрузки массива спецификаций.

Если при выполнении блока 2 обнаружены ошибки, вызванные отсутствием записей в массиве предметов, далее выполняются блоки 3, 4 и 5 после них — блоки 6 и 7. Если указанные ошибки отсутствуют, после блока 2 сразу выполняются блоки 6 и 7.

В блоках 3, 4 и 5 записи для пополнения массива предметов сортируются по коду предмета, исключаются записи с одинаковыми кодами (кроме первой) и производится добавление их в массив предметов. Включаемые в массив предметов записи печатаются в виде контрольной табуляграммы на устройстве SYS012.

В блоке 6 записи спецификаций сортируются в порядке возрастания кодов сборочной единицы (узла) и кода компоненты. В блоке 7 полученные после сортировки записи загружаются в массив спецификаций. При загрузке каждой записи устанавливаются двусторонние адресные связи между данной записью и записями узлов и компонент в массиве предметов, а также между соседними записями спецификаций данного узла. Загружаемые записи выводятся в контрольную табуляграмму на устройство SYS012 (для каждой сборочной единицы вывод начинается с нового листа). В конце загрузки на SYSLST печатаются характеристики массивов спецификаций и предметов, а на SYSLOG выдается сообщение BAA1001. Для загрузки 1000 записей, включая печать контрольной табуляграммы, требуется около 10 мин.

Во время загрузки массива спецификаций может произойти прерывание программы (например, из-за машинного сбоя). Так как к моменту прерывания часть спецификаций могла быть уже загружена, исправить положение повторным выполнением блока 7 нельзя, поскольку между массивами предметов и спецификаций

для части данных установлены адресные ссылки. Для повторной загрузки массива спецификаций необходимо иметь массив предметов в его исходном (первоначальном) состоянии. Для этой цели рекомендуется хранить копию массива предметов на магнитной ленте (копию можно получить после загрузки массива предметов с помощью программы CRDT) и восстанавливать его перед повторением загрузки массива спецификаций. После восстановления массива предметов для загрузки массива спецификаций необходимо выполнить блоки 5 и 7 (если в блоке 2 не были зафиксированы дополнения к массиву предметов, достаточно выполнить только блок 7).

После завершения загрузки на основании контрольных табуляграмм и табуляграмм ошибок необходимо произвести анализ полученных результатов и при необходимости внести изменения в массив банка данных.

Комплекс программ «Создание массива технологических маршрутов» предназначен для загрузки в массив технологических маршрутов информации формы № 01. Одновременно проводится проверка на полноту массивов предметов и рабочих мест и дополнение их недостающими записями.

Перечень программ комплекса приведен в табл. 44.

Таблица 44

Программа	Имя программы	Объем памяти, байт
Контроль исходных данных	AA501	5 000
Подготовка данных для загрузки массива технологических маршрутов и изменения базовых массивов	AA509	25 000
Сортировка изменений	AA203	13 000
Уплотнение изменений	AA506	6 000
Изменение массивов банка данных	AA507	35 000
Сортировка	AA202	13 000
Загрузка массива технологических маршрутов	AA510	35 000

Массив технологических маршрутов, как и массив спецификаций, является соединительным: записи массива технологических маршрутов связаны с записями деталей и сборочных единиц из массива предметов и записями оборудования из массива рабочих мест. Отсюда следует, что для загрузки массива технологических маршрутов требуется наличие обоих базовых массивов: предметов и рабочих мест.

В блоке 1 (рис. 14) производится ввод исходных данных с перфокарт, контроль правильности перфорации и запись проверенных данных на магнитную ленту. В блоке 2 проверенные данные считываются с магнитной ленты и проверяется наличие соответствующих записей в массиве предметов и в массиве рабочих мест. Если требуемая запись в базовом массиве отсутствует, она создается

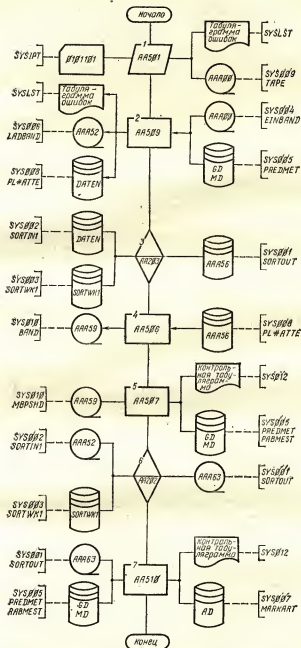
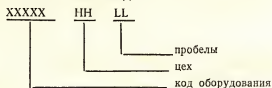


Рис. 14. Принципиальная блок-схема выполнения комплекса «Создание массива технологических маршрутов AD (Arbeitsplandatei)»

этим блоком. Для массива предметов запись состоит из кода предмета и символов XXXXX в поле «Наименование». Для массива рабочих мест запись содержит код рабочего места и символы XXXXX в поле «Наименование группы оборудования». Код рабочего места состоит из трех частей и имеет вид:



Для записей рабочих операций с номерами 990 и 999 создается искусственная запись (для массива рабочих мест), в котором код оборудования имеет значение 00000.

В незанятые поля в искусственных записях записываются пробелы. Созданные данным блоком искусственные записи выводятся на магнитный диск в массив изменений для базовых массивов и выдаются в виде табуляграммы ошибок на устройстве SYSLST. Записи рабочих операций выводятся на магнитную ленту и используются в дальнейшем для загрузки массива технологических маршрутов.

Если при выполнении блока 2 имели место ошибки, вызванные отсутствием записей в базовых массивах, дальше должны быть выполнены блоки 3, 4 и 5, после них — блоки 6 и 7. Если указанные ошибки отсутствовали, выполнение блоков 3, 4 и 5 не требуется. Их назначение аналогично соответствующим блокам комплекса «Создание массива спецификаций».

В блоке 6 записи рабочих операций технологических маршрутов сортируются по возрастанию кода предмета (детали, сборочной единицы) и номера рабочей операции.

Рассортированные записи в блоке 7 считываются с магнитной ленты и загружаются в массив технологических маршрутов. Запись первой рабочей операции технологического маршрута обработки предмета связывается с помощью адресных ссылок с соответствующей записью в массиве предметов. Все записи рабочих операций технологического маршрута по каждой детали связываются с соответствующими записями в массиве рабочих мест и друг с другом и выводятся в виде контрольной табуляграммы загрузки на устройство SYS012. Каждый технологический маршрут в табуляграмме распечатывается с нового листа. Загрузка заканчивается выводом характеристик массива технологических маршрутов и базовых массивов на устройство SYS1ST и сообщения BAA1001 — на SYSLOG.

Для загрузки 1000 записей, включая вывод контрольной табуляграммы, требуется около 10 мин.

Действия, выполняемые при прерываниях программы, аналогичны действиям, описанным при создании массива спецификаций, но при этом требуется восстановить оба базовых массива.

Комплекс программ «Изменение массивов банка данных» предназначен для актуализации данных. Функции изменения, выполняемые программами комплекса, приведены в табл. 35. Перечень программ комплекса дается в табл. 45.

Таблица 45

Программа	Имя программы	Объем памяти, байт
Контроль исходных данных	AA501	5 000
Предварительная сортировка	AA204	13 000
Подготовка изменений (макеты № 04, 07, 11 и 15)	AA511	23 000
Сортировка изменений	AA203	13 000
Уплотнение изменений	AA506	6 000
Изменение массивов банка	AA507	35 000
Подготовка изменений (макеты № 01—03, 05, 06, 08—10, 13 и 14)	AA512	23 000
Распределение групп материалов	AA513	15 000

Комплекс программ «Изменение массивов банка данных» обладает рядом особенностей:

для каждого макета при проведении изменений допустимыми являются только те функции, которые отмечены в табл. 36 знаком «+»;

в одном задании невозможно провести изменения для всех макетов. Допускается объединять в один пакет перфокарты макетов № 04, 07, 11 и 15 с проведением изменений согласно схеме, представленной на рис. 15, и остальных макетов с использованием схемы рис. 16;

функция «замена» (признак 5) выполняется отдельно, в то время как остальные функции (с признаком 0—4) могут выполняться в одном задании;

при добавлении уже существующих записей функция «добавление» выполняется как «корректировка»;

при отсутствии соответствующей записи в массиве функция «корректировка» выполняется как «добавление»;

если требуется стереть базовую запись, необходимо стереть и связанные с ней записи соединительных массивов, в противном случае будет выдано сообщение об ошибке, и стирание базовой записи не произойдет;

для стирания подетальных норм расхода основных материалов необходимо выполнить стирание записи соответствующей детали в массиве предметов, однако, если запись детали окажется связанной (например, для детали имеется технологический маршрут), подетальные нормы будут стерты, а сама запись детали останется.

Как видно из принципиальной блок-схемы (см. рис. 15 и 16), изменения для обеих групп макетов выполняются примерно одинаково. Программа AA501 вводит данные с перфокарт, проверяет правильность их перфорации и записывает на магнитную ленту.

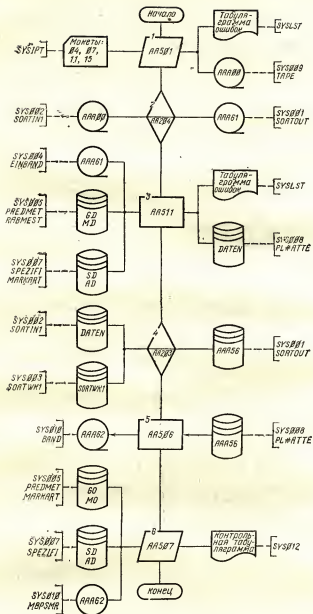


Рис. 15. Принципиальная блок-схема выполнения комплекса «Изменение массивов банка данных» (для макетов № 04, 07, 11, 15)

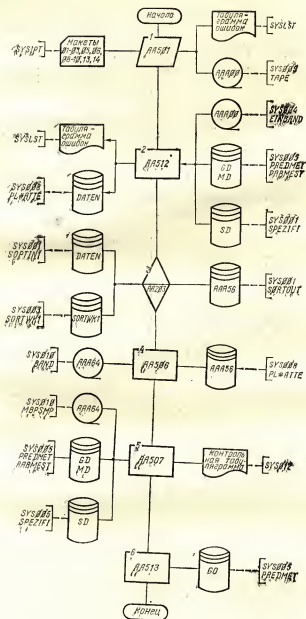


Рис. 16. Принципиальная блок-схема выполнения комплекса «Изменение массивов банка данных» (для макетов № 01—03, 05, 06, 08—10, 13 и 14)

Программа AA204 (см. рис. 16) производит предварительную сортировку данных (это требуется только для макета № 07). Программа AA511 преобразует входные записи к виду, удобному для проведения необходимых изменений. Здесь же могут создаваться искусственные записи для добавления в базовые массивы. Эти записи содержат в полях «Наименование» символы XXXXX и *** и должны быть в дальнейшем скорректированы (с помощью функции 3).

Программа AA203 сортирует обработанные данные, после чего программа AA506 формирует массив изменений путем удаления повторяющихся записей. Программа AA507 вносит изменения в массивы банка данных.

Программа AA513 (см. рис. 16) помещает код групп материалов в записи материалов из массива предметов. Этот блок необходимо выполнять в случае, если проводились изменения с помощью макетов № 02, 03 или 14. Если во время выполнения программы AA507 происходит прерывание (например, из-за сбоя машины), для продолжения работы необходимо сначала восстановить все массивы банка данных и выполнить эту программу заново.

При выполнении изменений на устройство SYS012 выдается контрольная табуляграмма изменений. Для базовых массивов печатаются все изменяемые записи с указанием выполненных функций. Для соединительных массивов печатывается целиком спецификация или технологический маршрут. Строки, которые подвергались изменениям, имеют сокращенное обозначение функции, остальные строки распечатываются без отметок об изменении.

Программы выполняются в режиме пакетной обработки под управлением ДОС ЕС. Пакет заданий включает управляющие операторы программы управления заданиями и управляющие операторы рабочих программ.

При формировании пакета заданий в него необходимо включить управляющие операторы для постоянных назначений устройств, требуемых в процессе работы с системой БАСТАИ, и для записи меток в область постоянной информации о метках. Чтобы облегчить формирование пакета заданий, для каждой программы приводится набор управляющих операторов с указанием в операторе //JOB, к какой программе они относятся. Из управляющих операторов, отмеченных цифрой со скобкой, для каждого комплекса требуется только один оператор. Цифра со скобкой указывает номер блок-схемы комплекса, для которого предназначен этот оператор.

На блок-схеме (см. рис. 11—16) содержатся все данные, необходимые для формирования пакетов заданий.

Набор управляющих операторов включает следующие карты:
а) для выполнения постоянных назначений устройств

```
// OPTION PARSTD  
ASSGN SYSSLB,X'191'  
ASSGN SYSRLB,X'191'  
ASSGN SYS003,X'191'
```



```

ASSGN SYS004,X'282'
ASSGN SYS005,X'192'
ASSGN SYS006,X'281'
ASSGN SYS007,X'193'
ASSGN SYS008,X'191'
ASSGN SYS010,X'280'
ASSGN SYS012,X'00F'
// OPTION PARSTD BG
// DLBL IJSYSS4,'SOPS/BA/QUE',99/365
// EXTENT SYSSLB,BASTEI,1,0,10,500
// DLBL IJSYRL,'SOPS/BA/OBJ',99/365
// EXTENT SYSRLB,1,0,510,250
// DLBL PREDMET,'GEGENSTANDSDATEI',99/365,DA
// EXTENT SYS005,BANKB1,1,0,10,2
// EXTENT SYS006,BANKB1,1,1,12,18
// EXTENT SYS005,BANKB1,1,2,30,1250
// DLBL SPEZIFI,'STUECKLISTENDATEI',99/365,DA
// EXTENT SYS007,BANKB2,1,0,10,550
// DLBL RABMEST,'ARBEITSPLATZDATEI',99/365,DA
// EXTENT SYS005,BANKB1,1,0,1280,2
// EXTENT SYS005,BANKB1,1,1,1282,18
// EXTENT SYS005,BANKB1,1,2,1300,560
// DLBL MARKART,'ARBEITSPLANDATEI',99/365,DA
// EXTENT SYS007,BANKB2,1,0,560,1100
// DLBL SORTWK1,03/365
// EXTENT SYS003,BASTEI,1,0,1170,420
// DLBL PL ATTE,'DATEN',03/365
// EXTENT SYS008,BASTEI,1,0,760,410

```

б) для выполнения программ

```

// JOB AA501
// OPTION LINK
// INCLUDE CONTROL
// LBLTYR TAPE
// EXEC LNKEDT
// ASSGN SYS009,X'283'11)
// ASSGN SYS009,X'282'12-16)
// MTC REW, SYS009
// TLBL TAPE,'AAA00'
// EXEC
/&

// JOB AA201
// ASSGN SYS002,X'283'
// ASSGN SYS001,X'282'
// MTC REW, SYS002
// TLBL SORTINI,'AAA00'
// TLBL SORTOUT,'AAA51'
// LBLTYP TAPE
// EXEC SORT
SORT FIELDS=(4,13,B1A),WORK=1
RECORD TYPE=F,LENGTH=30
INFIL BLKSIZE=1600,BYPASS, OPEN=NORWD
OUTFIL BLKSIZE=1600,OPEN=RWD
OPTION LABEL=(S,S,U)
END
/*
/&

// JOB AA502
// OPTION LINK

```

```

        INCLUDE B#EPLAD1
// LBLTYP NSD(03)
// EXEC LNKEDT
// MTC REW,SYS004
// MTC REW,SYS006
// TLBL EINBAND,'AAA51'
// TLBL LADBAND,'AAA52'
// EXEC
/&

// JOB AA202
// ASSGN SYS001,X'280'
// ASSGN SYS002,X'281'
// MTC REW,SYS002
// TLBL SORTOUT,'AAA53'11)
// TLBL SORTOUT,'AAA54'12)
// TLBL SORTOUT,'AAA58'13)
// TLBL SORTOUT,'AAA60'14)
// TLBL SORTIN,'AAA52'
// LBLTYP TAPE
// EXEC SORT
        SORT FIELDS=(14,30,CH,A),WORK=1,SIZE=2000
        REKORD TYPE=V,LENGTH (90,,90,90)
        INPFIL BLKSIZE=1804,BYPASS,OPEN=NORWD
        OUTFIL BLKSIZE=1804
        OPTION LABEL=(S,S,U)
        END
/&

// JOB AA503
// OPTION LINK
        INCLUDE B#LEITLA
        INCLUDE B#DORGGD
// LBLTYP MSD(03)
// EXEC LNKEDT
// MTC REW,SYS010
// TLBL MBPSMR,'AAA53'
/&

// JOB AA504
// OPTION LINK
        INCLUDE B#LEITLA
        INCLUDE B#DORGMD
// LBLTYP NSD(03)
// EXEC LNKEDT
// MTC REW,SYS010
// TLBL MBPSMR,'AA54'
// EXEC
/&

// JOB AA505
// OPTION LINK
        INCLUDE B#EPLAD2
        INCLUDE B#WURZEL
        INCLUDE B#ARBEIT
        INCLUDE B#EROABS
// LBLTYP NSD(03)
// EXEC LNKEDT
// MTC REW,SYS004
// TLBL EINBAND,'AAA00'
// MTC REW,SYS006

```

```
// TLBL LADBAND,'AAA52'  
// EXEC  
/&
```

```
// JOB AA203  
// ASSGN SYS001,X'191'  
// ASSGN SYS002,X'191'  
// DLBL SORTOUT,'AAA56'  
// EXTENT SYS001,BASTE1,1,0,1590,410  
// DLBL SORTIN  
// EXTENT,SYS002,BASTE1,1,0,760,41  
// EXEC SORT  
SORT FIELDS=(11,35,CH,A),WORK=1,SIZE=2500  
REKORD TYPE=F,LENGTH=(90,90)  
INPFIL BYPASS  
OPTION LABEL=(U,U,U;  
END  
/&
```

```
// JOB AA506  
// OPTION LINK  
INCLUDE B#ZUSAM7  
// LBLTYP NSD(03)  
// MTC REW,SYS010  
// TLBL BAND,'AAA57'13)  
// DLBL PL#ATTE,'AAA57'  
// EXTENT SYS008,BASTE1,1,0,1590,410  
// TLBL BAND,'AAA59'14)  
// TLBL BAND,'AAA62'15)  
// TLBL BAND,'AAA63'16)  
// EXEC  
// MTC REW,SYS010  
/&
```

```
// JOB AA507  
// OPTION LINK  
INCLUDE B#LEITLA  
INCLUDE B#DORGAE  
INCLUDE B#WURZEL  
INCLUDE B#ARBEIT  
INCLUDE B#EROABS  
// LBLTYP NSD(03)  
// EXEC LNKEDT  
// MTC REW,SYS010  
// TLBL MBPSMR,'AAA57'13)  
// TLBL MBPSMR,'AAA59'14)  
// TLBL MBPSMR,'AAA62'15)  
// TLBL MBPSMR,'AAA63'16)  
// EXEC  
// MTC REW,SYS010  
/&
```

```
// JOB AA508  
// OPTION LINK  
INCLUDE B#LEITLA  
INCLUDE B#DORGSD  
INCLUDE B#WURZEL  
INCLUDE B#ARBEIT  
INCLUDE B#EROABS
```

```

// LBLTYP NSD(03)
// EXEC LNKEDT
// MTC REW,SYS010
// TLBL MBPSMK,'AAA58'
// EXEC
/&

// JOB AA509
// OPTION LINK
// INCLUDE B#EPLAD3
// INCLUDE B#WURZEL
// INCLUDE B#ARBEIT
// INCLUDE B#EROABS
// LBLTYP NSD(03)
// EXEC LNKEDT
// DLBL PL#ATTE
// EXTENT SYS008,BASTEI,1,0, 760,600
// MTC REW,SYS004
// MTC REW,SYS006
// TLBL EINBAND,'AAA00'
// TLBL LADBAND,'AAA52'
// EXEC
/&

// JOB AA510
// OPTION LINK
// INCLUDE B#LEITLA
// INCLUDE B#DORGAD
// INCLUDE B#WURZEL
// INCLUDE B#ARBEIT
// INCLUDE B#EROABS
// LBLTYP NSD(03)
// EXEC LNKEDT
// MTC REW,SYS010
// TLBL MBPSMR,'AAA60'
// EXEC
/&

// JOB AA204
// ASSGN SYS001,X'280'
// ASSGN SYS002,X'281'
// TLBL SORTIN1,'AAA00'
// TLBL SORTOUT,'AAA61'
// LBLTYP TAPE
// EXEC SORT
// SORT FIELDS=(1,16,B1,22,13,B1,A),WORK=1
// RECORD TYPE=F,LENGTH=80
// INPFIL BLKSIZE=1600,BYPASS
// OUTFIL BLKSIZE=1600
// OPTION LABEL=(S,S,U)
// END
/*
/&

// JOB AA511
// UPSI 11
// OPTION LINK
// INCLUDE B#EPVAR
// INCLUDE B#WURZEL
// INCLUDE B#ARBEIT
// INCLUDE B#EROABS

```

```
// LBLTYP NSD(03)
// EXEC LNKEDT
// ASSGN SYS004,X'282'
// MTC REW,SYS004
// TLBL EINBAND,'AAA04'
// EXEC
/&
```

```
// JOB AA512
// OPTION LINK
// UPSI 11
// INCLUDE B#EPAEND
// INCLUDE B#WURZEL
// INCLUDE B#ARBEIT
// INCLUDE B#EROABS
// LBLTYP NSD(03)
// EXEC LNKEDT
// ASSGN SYS004,X'282'
// MTC REW,SYS004
// TLBL EINBAND,'AAA00'
// EXEC
/&
```

```
// JOB AA513
// OPTION LINK
// INCLUDE B#GRUVER
// INCLUDE B#WURZEL
// INCLUDE B#ARBEIT
// INCLUDE B#EROABS
// LBLTYR NSD(03)
// EXEC LNKEDT
// EXEC
/&
```

Пакет заданий для выполнения комплекса «Создание массива предметов» komponуется из управляющих операторов для программ: AA501, AA201, AA502, AA202, AA503 (см. рис. 11). Массив AAA53 в процессе загрузки не изменяется и для повторного создания массива предметов достаточно выполнить только программу AA503 (при условии, что массив AAA53 сохранился).

Пакет задания для выполнения комплекса «Создание массива рабочих мест» (см. рис. 12) komponуется из управляющих операторов для программ: AA501, AA502, AA202, AA504. Для выполнения повторной загрузки массива рабочих мест достаточно включить в пакет только управляющие операторы программы AA504 (при условии, что массив AAA54 сохранился).

Пакет задания для выполнения комплекса «Создание массива спецификаций» (см. рис. 13) komponуется из управляющих операторов программ: AA501, AA505, AA203, AA506, AA507, AA202, AA508. Если комплекс в предыдущий раз не был закончен из-за сбоя при выполнении программы AA507, необходимо сначала заново загрузить массив предметов или восстановить его с магнитной ленты, для чего необходимо иметь копию массива предметов после его последней корректировки, и после этого выполнить прог-

раммы AA507, AA202 и AA508. При сбое во время выполнения программы AA508 нужно поступать аналогичным образом, т. е. восстановить массив предметов и продолжить выполнение с программы AA507.

Если при выполнении программы AA505 в табуляграмме ошибок нет сообщений об отсутствии необходимых записей в массиве предметов, программы AA203, AA506 и AA507 из пакета можно исключить.

Пакет заданий для создания массива технологических маршрутов komponуется из управляющих операторов программ AA501, AA509, AA203, AA506, AA507, AA202, AA510 (см. рис. 14). Перед выполнением данного пакета необходимо получить копию ранее загруженных массивов предметов, рабочих мест и спецификаций для экономии времени, если необходима повторная загрузка массива технологических маршрутов из-за возможных сбоев машины при выполнении программы AA507 или AA510.

При формировании пакета для выполнения комплекса «Изменение массива банков данных» необходимо учитывать следующие особенности:

представленные на рис. 15 и 16 схемы предназначены для обработки двух различных групп входных документов. По первой схеме формируется пакет для обработки макетов № 04, 07, 11 и 15, по второй — остальные макеты;

в одном пакете заданий могут обрабатываться входные данные либо только с функциями 0—4, либо только с функцией 5 (замена);

для повторного проведения одних и тех же изменений (такая необходимость может возникнуть, если при выполнении программы AA507 произошел неустраняемый сбой машины) рекомендуется перед внесением изменений получить копию всех массивов банка данных. Целесообразно получать такие копии периодически (например, ежемесячно) или после проведения определенного объема изменений.

Сформированный пакет заданий передается оператору вместе с бланком заказа на выполнение задания, в котором указываются используемые устройства, носители информации, их маркировка, возможные варианты замены.

Перед выполнением заданий оператор обязан:

- а) проверить наличие управляющих карт и требуемых носителей информации согласно бланку заказа на выполнение;
- б) проверить готовность используемых внешних устройств;
- в) при необходимости произвести замену назначений для внешних устройств, если это позволяет комплектность ЭВМ.

Для выполнения задания необходимо заправить пакет управляющих карт в устройство ввода и нажать кнопку «Пуск».

При машинном или программном сбое следует повторить выполнение прерванного задания (или шага задания). Если сбой произошел во время выполнения одной из программ AA503, AA504,

AA507, AA508, AA510, нужно прекратить дальнейшее выполнение пакета, снять и сохранить магнитные ленты с входными для этих программ массивами (при сбое в программе AA507 нужно дополнительно сохранить массив AAA52 (см. рис. 13 и 14). Если для повторения требуется первоначальная загрузка, необходимо выполнить пакет заданий для постоянных назначений. При повторном сбое задание возвращается заказчику со всеми полученными результатами и объяснением причин невыполнения задания.

Информация, сопровождающая выполнение программ и не требующая вмешательства оператора, приведена в табл. 46.

Таблица 46

Номер сообщения	Текст сообщения. Описание сообщения
МЧ3310	<p>ПРИЗНАК ФУНКЦИИ ДЕФЕКТ</p> <p>Сообщение выдается программами подготовки В#ЕРАЕНД и В#ЕРВАР, если во входной записи встретился недопустимый признак функции. Входная запись распечатывается и пропускается</p>
МЧ3312	<p>XXXXXXXXXXXX СПЕЦИФИКАЦИИ НЕТ, КОМПЛЕКТОВОЧНАЯ КАРТА НЕ ЗАГРУЗ</p> <p>Сообщение выдается программой В#ЕРВАР, если вводится комплектовочная карта, а в банке данных для этого узла отсутствует спецификация. Распечатывается код узла, и входная запись пропускается</p>
МЧ3313	<p>УЗЕЛ XXXXXXXXXXXX ДЕТАЛЬ XXXXXXXXXXXX СПЕЦИФИКАЦИЯ И КОМПЛЕКТОВОЧНАЯ КАРТА НЕ СОВПАДАЮТ В СПЕЦ. (XX) И КОМПЛ. КАРТЕ (XX) НЕ СОВПАДАЕТ КОЛ-ВО</p> <p>Сообщение выдается программой В#ЕРВАР при вводе комплектовочной карты, если количество в комплектовочной карте не совпадает с количеством, введенным ранее из спецификаций. Комплектовочная карта вводится в банк данных и заменяет спецификацию</p>
МЧ3314	<p>УЗЕЛ XXXXXXXXXXXX ДЕТАЛЬ XXXXXXXXXXXX СПЕЦИФИКАЦИЯ И КОМПЛЕКТОВОЧНАЯ КАРТА НЕ СОВПАДАЮТ</p> <p>Сообщение выдается программами В#ЕРВАР и В#ЕРАЕНД, если при вводе комплектовочной карты обнаружены компоненты, отличные от введенных ранее компонент из спецификаций. Комплектовочная карта вводится и заменяет спецификацию</p>
МЧ3315	<p>XXXXXXXXXXXX КОМПЛЕКТОВОЧНАЯ УЖЕ ЕСТЬ</p> <p>Поступила входная запись на изменение спецификаций, а в банк данных уже раньше была введена комплектовочная карта на данный узел. Входная запись спецификаций не обрабатывается (пропускается)</p>

Номер сообщения	Текст сообщения. Описание сообщения
МЧ3316	<p>XXXXXXXXXXXX СПЕЦИФИКАЦИЙ НЕТ</p> <p>Поступила входная запись на замену кода узла, а спецификация или комплектовочная карта для данного узла еще не вводились. Входная запись пропускается</p>
МЧ3317	<p>XXXXXXXXXXXX ПОДЕТАЛЬНЫХ НОРМ НЕТ</p> <p>Поступила запись на корректировку подетальных норм расхода материалов, а норма расхода на данную деталь раньше не вводилась. Входная запись пропускается</p>
МЧ3318	<p>XXXXXXXXXXXX НЕТ ТЕХНОЛОГИЧЕСКОГО МАРШРУТА</p> <p>Поступила входная запись для замены кода предмета в технологическом маршруте, а технологический маршрут для данного предмета еще не введен. Входная запись игнорируется</p>
МЧ3323	<p>КОД УЗЛА XXXXXXXXXXXX ОТСУТСТВУЕТ В МАССИВЕ ПРЕДМЕТОВ</p> <p>Для подлежащей загрузке спецификации в массиве предметов отсутствует код узла. Создается искусственная запись для добавления в МП, содержащая отметку XXXXX в поле «Название»</p>
МЧ3324	<p>КОД КОМП. XXXXXXXXXXXX ОТСУТСТВУЕТ В МАССИВЕ ПРЕДМЕТОВ, КОД УЗЛА XXXXXXXXXXXX</p> <p>Для подлежащей загрузке спецификации в МП отсутствует код компоненты. Создается искусственная запись для добавления в МП, содержащая отметку XXXXX в поле «Название»</p>
МЧ3330	<p>XXXX НННННН</p> <p>При считывании записи из банка данных произошла ошибка считывания (сбой диска). Символы XXXX отражают значение байт ошибок в префиксе рабочей области. НН...Н — имя массива, при чтении которого произошел сбой. Если эта ошибка повторяется, необходимо восстановить диск и выполнить программу повторно</p>

§ 4. Расчет применяемости деталей и сборочных единиц в изделии

Общее описание. Задача «Расчет применяемости» предназначена для формирования информации о применяемости деталей и сборочных единиц в рассчитываемом изделии, т. е. информации о номенклатуре и количестве всех входящих в отдельное изделие составных частей. Рассчитываемым изделием может быть не только конечное изделие, но и любая сборочная единица, о структурном и количественном составе которой необходимо получить информацию.

Расчет осуществляется на основе исходных данных, содержащихся в конструкторских спецификациях, и выполняется в двух вариантах:

первый вариант — по ограниченной номенклатуре изделий (сборочных единиц);

второй вариант — по всей номенклатуре конечных изделий и сборочных единиц, информация о которых содержится в исходных данных.

В результате решения выдаются табуляграммы применяемости по заданному изделию или по всей номенклатуре изделий (формы № 13 и 14).

Форма № 13

ВЕДОМОСТЬ ПРИМЕНЯЕМОСТИ ДЕТАЛЕЙ И СБОРОЧНЫХ ЕДИНИЦ В ИЗДЕЛИИ							
..... 197 .. г.							
Что входит			Куда аходит			Количество	
Краткое наименование детали (сборочной единицы)	Код детали (сборочной единицы)	Код варианта исполнения	Краткое наименование сборочной единицы	Код сборочной единицы	Код варианта исполнения	На сборочную единицу	На изделие
1	2	3	4	5	6	7	8
Краткое наименование изделия			Код изделия		Код варианта исполнения		

Форма № 14

ВЕДОМОСТЬ ПРИМЕНЯЕМОСТИ ДЕТАЛИ, СБОРОЧНОЙ ЕДИНИЦЫ В ИЗДЕЛИЯХ, СБОРОЧНЫХ ЕДИНИЦАХ						
..... 197 .. г.						
Что аходит			Куда аходит			Общее количество
Краткое наименование детали (сборочной единицы)	Код детали (сборочной единицы)	Код варианта исполнения	Краткое наименование изделия (сборочной единицы)	Код изделия (сборочной единицы)	Код варианта исполнения	
1	2	3	4	5	6	7

Исходными данными для решения задачи являются сведения документа «Список изделий», в котором указываются коды изделий (сборочных единиц) и вариант их исполнения, по которым необходимо выполнить расчет применяемости, и «Спецификация изделий». Первый из документов заполняется в случае выполнения расчета применяемости составных частей в изделиях, перечень которых ограничен, — первый вариант расчета.

При решении по 2-му варианту вместо карт с кодами изделий используется заранее подготовленная перфокарта со специальным признаком. Входной информацией в этом случае является массив

«Спецификация изделий», содержащая сведения о коде изделия, коде сборочной единицы (куда входит), коде детали, сборочной единицы (что входит), количестве. Здесь и далее при описании задач дается перечень используемых реквизитов ранее приведенных массивов банка данных и полученных реквизитов выходных массивов, с тем чтобы облегчить восприятие принципиальной сущности задач.

Алгоритм решения задач построен на основании реально существующей структуры массивов банка данных, что иллюстрируют приводимые для каждой задачи принципиальные блок-схемы их решения и общая принципиальная схема информационной увязки рассматриваемых задач (рис. 17). Аналогичным образом приводятся сведения о выходных массивах. В данном случае результатом решения задачи является выходной массив, содержащий сведения, представленные в табл. 47.

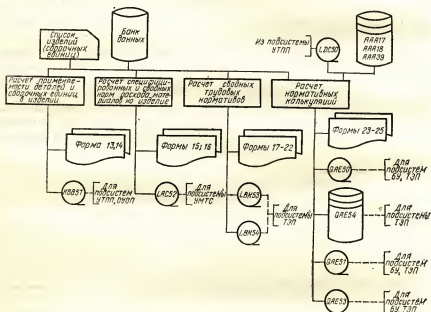


Рис. 17. Схема информационной увязки задач: УТПП — подсистема управления технической подготовкой производства; ОУОП — подсистема оперативного управления основным производством; УМТС — подсистема управления материально-техническим снабжением; ТЭП — подсистема технико-экономического планирования; БУ — подсистема управления бухгалтерским учетом

Алгоритм решения задачи. При решении задачи по первому варианту рассчитывается состав только тех изделий, коды которых включены в документ «Список изделий». При выполнении расчета по второму варианту кодами рассчитываемых изделий являются

Таблица 47

Реквизит	Длина реквизита, знаки
Код изделия	9(11)
Код сборочной единицы (куда входит)	9(11)
Код сборочной единицы (что входит)	9(11)
Наименование (что входит)	A(20)
Наименование (куда входит)	A(20)
Количество на сборочную единицу	9(2)
Количество на изделие	9(2)
Степень вхождения	9(2)

коды всех сборочных единиц (куда входит), записанные в массив предметов банка данных.

По каждому коду рассчитываемого изделия, которому приписывается наименьшая, нулевая степень вхождения, выполняется набор входящих в изделие составных частей и расчет количественных показателей.

Набор состава изделия выполняется по логической схеме «куда входит — что входит — куда входит —...» путем оперирования зафиксированными в конструкторской спецификации адресными связями каждой сборочной единицы с непосредственно входящими в нее составными частями. Одновременно с набором состава изделия фиксируется степень вхождения и производится расчет количества, в котором составная часть применяется в изделии по формуле

$$P_{i+1} = P_i \cdot S_{i+1},$$

где P_i — количество на изделие, вычисленное на предыдущей степени i (P_0 принимается равным 1);

S_{i+1} — количество на сборочную единицу из спецификации, определяемое на $(i+1)$ -й степени расчета ($i=1, 2, 3, \dots, t$).

По каждой составной части изделия в выходную информацию включаются, помимо расчетных данных, взятые из спецификации первичные сведения о ее непосредственной входимости (значения реквизитов «куда входит» и «количество на сборочную единицу»).

Если при выдаче табуляграммы (см. форму № 13) об изделии имеется несколько записей с одинаковыми значениями реквизитов «что входит», то производится суммирование значений реквизитов «количество на изделие» по формуле

$$P = \sum_{i=1}^t P_{i+1},$$

где P — общее количество составных частей данного кода в изделии.

Сумма записывается в графу «Количество на изделие» под итоговой чертой за последней документострокой из группы строк с одинаковыми кодами «что входит».

Алгоритм решения задачи на ЭВМ реализуется посредством выполнения трех программ:

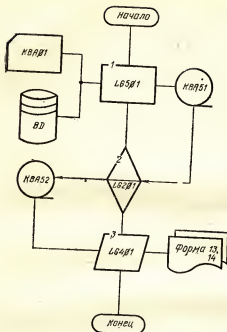


Рис. 18. Принципиальная блок-схема решения задачи «Расчет применимости деталей и сборочных единиц в изделии»

структурного разузло-
вания состава изделий;
сортировки массива при-
меняемости деталей и сбор-
очных единиц в изделиях;
вывода на печать табуля-
граммы «Ведомость при-
меняемости деталей, сбороч-
ных единиц в изделии».

Как видно из рис. 18, технологическая схема реше-
ния задачи реализована тре-
мя программами: LG501 —
формирование массива при-
меняемости; LG201 — сорти-
ровка массива применяемо-
сти; LG401 — печать табуля-
грамм (см. формы № 13 и 14).

С помощью программы LG501 осуществляется ввод оперативной информации с перфокарт и формирование массива КВА51.

При решении задачи по первому варианту (по ограни-
ченному списку) перечень кодов рассчитываемых изде-
лий перфорируется по схеме, представленной в табл. 48.

Таблица 48

Реквизит	Номер колонок на ПК	Шаблон
Макет (20)	1—2	9(2)
Код изделия	3—13	9(11)
Код варианта исполнения	14—15	9(2)

При решении по второму варианту перфорируется одна перфокарта с символом «V» в третьей колонке.

В процессе формирования массива КВА51 строятся записи двух типов: шапка и строка. При сортировке записи типа шапка оказываются впереди записей типа строка, так как не занятые в записях первого типа поля содержат пробел. При печати табуляграммы из записей первого типа формируется шапка табуляграммы, а записи второго типа преобразуются в строки табуляграмм. В табл. 49 приведена структура массива КВА51.

Таблица 49

Реквизит	П. абз.он	Тип реквизита	Примечание*
Код изделия	9(11)	Z	1
Что входит (код детали, сборочной единицы)	9(11)	Z	2
Куда входит (код детали, сборочной единицы)	9(11)	Z	3
Количество на сборочную единицу	9(2)	P	4
Количество на изделие	9(2)	P	
Степень вхождения	9(2)	Z	
Наименование «что входит»	A(20)	C	
Наименование изделия	A(20)	C	
Наименование «куда входит»	A(20)	C	
Код варианта исполнения	9(2)	Z	

* Цифрами в примечании отмечены ключевые реквизиты и их старшинство при упорядочивании массива КВА51.

§ 5. Расчет специфицированных и сводных норм расхода материалов

Общее описание. Данная задача предназначена для расчета специфицированных норм расхода каждого материала на изделие по предприятию, а также норм расхода материалов в укрупненной номенклатуре на изделие по предприятию. В результате решения задачи определяются следующие показатели:

масса на 1000 изделий в специфицированной номенклатуре по предприятию в подетальном разрезе;

норма расхода на 1000 изделий в специфицированной номенклатуре по предприятию в подетальном разрезе;

коэффициент использования материала в специфицированной номенклатуре по предприятию;

масса на 1000 изделий в укрупненной номенклатуре по предприятию;

норма расхода на 1000 изделий в укрупненной номенклатуре по предприятию;

коэффициент использования групп материалов по предприятию.

Периодичность решения задачи определяется в зависимости от потребности в данной информации на предприятии, от периодичности решения других задач, использующих информацию данной задачи.

Результаты решения задачи используются службами предприятия для учета и анализа использования материалов; для расчета потребности в материалах в специфицированной номенклатуре и укрупненной номенклатуре на плановый период.

Исходными данными для решения задачи являются следующие сведения документов «Список изделий», «Спецификация изделий» и «Справочник подетальных норм расхода основных материалов»:

код детали, код цеха, код материала, код единицы измерения, норма расхода материала, масса заготовки, чистый вес.

В результате решения задачи формируется массив «Специфицированные нормы расхода основных материалов на изделие, сборочную единицу», используемый при решении других задач. Состав и описание реквизитов массива приведены в табл. 50.

Таблица 50

Реквизит	Длина реквизита, знаки
Код изделия (сборочной единицы)	9(11)
Код материала	9(8)
Код единицы измерения	9(2)
Код цеха	9(2)
Норма расхода	9(4), 9(4)
Код единицы нормирования	9(1)

Кроме того, выдаются на печать табуляграммы «Подетально-специфицированные нормы расхода основных материалов» (форма № 15) и «Сводные нормы расхода основных материалов на изделие» (форма № 16).

Форма № 15

ПОДЕТАЛЬНО-СПЕЦИФИЦИРОВАННЫЕ НОРМЫ РАСХОДА ОСНОВНЫХ МАТЕРИАЛОВ "_____" 197 г.						Наименование изделия		
						Код изделия		
						Код варианта исполнения		
Деталь			Количество на сборочную единицу	Код материала	Код единицы измерения	На 1000 сборочных единиц		Коэффициент использования
Наименование	Код	Вариант исполнения				чистый вес	норма расхода	
1	2	3	4	5	6	7	8	9

Код группы. Наименование и характеристика материала

Итого:

Всего по группе:

× × ×
× × ×

Алгоритм решения задачи. Для выдачи на печать табуляграммы «Подетально-специфицированные нормы расхода основных материалов» реквизиты «масса M_{ijk} », «норма расхода N_{ijk} » и «коэффициент использования Q_k » рассчитываются с учетом коэффициента нормирования Q по следующим формулам:

СВОДНЫЕ НОРМЫ РАСХОДА ОСНОВНЫХ МАТЕРИАЛОВ НА ИЗДЕЛИЕ						Наименование изделия			
. 197__г.						Код изделия			
						Код варианта исполнения			
Номер соответ- ствия	Материал			Текущий 197__г.			Планируемый 197__г.		
	Наимено- вание сводной группы	Код группы	Код единицы измерения	Чистый вес	Норма расхода	Коэффициент использования	Чистый вес	Норма расхода	Коэффициент использования
				на 1000 изделий			на 1000 изделий		
	1	2	3	4	5	6	7	8	9

$$M_{ijh} = M_{ij} \cdot P_{jh} \cdot Q,$$

где M_{ij} — масса i -го материала на j -ю деталь ($i=1, 2, \dots, m, j=1, 2, \dots, n$);

P_{jh} — общее количество j -х деталей в h -м изделии;

$$N_{ijk} = N_{ij} \cdot P_{jh} \cdot Q,$$

где N_{ij} — норма расхода i -го материала на j -ю деталь;

$$Q_h = \frac{M_{ijk}}{N_{ijk}}.$$

Коэффициент нормирования Q имеет следующие значения: если в массиве PREDMET код единицы нормирования 8, то весовые показатели составлены на 10 деталей, а коэффициент нормирования $Q=100$; если код единицы нормирования 2, то весовые показатели составлены на 1000 деталей, а коэффициент нормирования $Q=1$.

Табуляграмма выдается с указанием группы материала по каждой детали, а также подводится итог материала в специфицированной номенклатуре в пределах группы и общий итог группы материала на изделие:

$$M_{ik}^s = \sum_{j=1}^n M_{ijk};$$

$$N_{ik}^s = \sum_{j=1}^n N_{ijk};$$

$$Q_k^s = \frac{M_{ik}^s}{N_{ik}^s}.$$

где M_{ik}^s — масса i -го материала s -й группы на k -е изделие ($s=1, 2, \dots, r$);

N_{ik}^s — норма расхода i -го материала s -й группы на k -е изделие;

Q_k^s — коэффициент использования s -й группы материала на k -е изделие.

Общий итог реквизитов «масса» и «норма расхода» по сводным группам материалов на изделие подводится по следующим формулам:

$$M_k^p = \sum_{i=1}^m \sum_{s=1}^r M_{ik}^s ;$$

$$N_k^p = \sum_{i=1}^m \sum_{s=1}^r N_{ik}^s ;$$

$$Q_k^p = \frac{M_k^p}{N_k^p} ,$$

где M_k^p — масса p -й сводной группы материала на k -е изделие ($p=1, 2, \dots, q$);

N_k^p — норма расхода p -й сводной группы материала на k -е изделие;

Q_k^p — коэффициент использования материала по p -й сводной группе на k -е изделие.

Реквизиты «наименования изделия», «код изделия», «код варианта исполнения», «наименование детали» выдаются на печать из массива PREDMET на основании массива «Список изделий, сборочных единиц». При этом последние два разряда кода изделия (детали) являются кодом варианта исполнения.

Реквизиты «код материала», «наименование материала», «код группы материала» и «код единицы измерения материала» выдаются на печать из массива предметов по каждой детали, входящей в изделие. Значение реквизита «количество на сборочную единицу» выводится на печать из массива SPECIFI по деталям, входящим в рассчитываемое изделие, а реквизиты «номер соответствия», «наименование группы материала» и «код единицы измерения группы материала» — из массива PREDMET по коду группы материала.

В табуляграмме «Сводные нормы расхода основных материалов на изделие» выдаются общие итоги массы и нормы расхода по группе материала на изделие.

Как видно из рис. 19, технологическая схема решения задачи реализована шестью программами:

LA501 — полное разузлование спецификаций;

LA201 — сортировка массива полного разузлования спецификаций;

LA401 — выдача на печать формы № 15 и формирование хранимого массива;

LA202 — сортировка специфицированных норм расхода основных материалов;

LA203 — сортировка сводных норм расхода основных материалов;

LA402 — выдача на печать формы № 16.

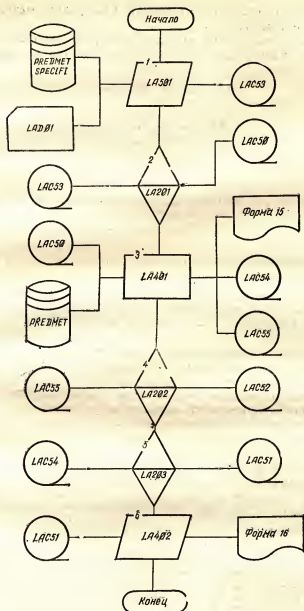


Рис. 19. Принципиальная блок-схема решения задачи «Расчет специфицированных и сводных норм расхода материалов»

С помощью программы LA501 осуществляется ввод оперативной информации с перфокарт и на основании информации массивов PREDMET и SPECIFI производится полное разузлование спецификаций. В результате формируется массив LAC53.

Записи рассортированного массива полного разузлования выводятся на печать (программа LA401) в виде формы № 15. При этом формируется массив специфицированных норм расхода основных материалов LAC55, который сортируется и хранится для подсистемы управления материально-техническим снабжением, и массив сводных норм расхода основных материалов LAC54, используемый после соответствующей сортировки (программа LA203) для выдачи формы № 16. Структура массива LAC52 представлена в табл. 50.

При решении задачи по первому варианту (ограниченный список) перечень кодов рассчитываемых изделий перфорируется на перфокартах по схеме, представленной в табл. 51. При решении по второму варианту перфорируется одна перфокарта с символом «V» в третьей колонке.

Таблица 51

Реквизит	Номер колонок на перфокартах	Шаблон
Макет (20)	1—2	9(2)
Код изделия	3—13	9(11)
Код варианта исполнения	14—15	9(2)

В процессе формирования массива LAC53 строятся записи двух типов: шапка и строка. В результате сортировки (LA201) записи типа шапка предшествуют записи типа строка для одного и того же изделия.

§ 6. Расчет сводных трудовых нормативов

Общее описание. Задача «Расчет сводных трудовых нормативов» входит в подсистему технической подготовки производства и предназначена для расчета сводной нормативной трудоемкости и заработной платы на деталь, сборочную единицу и изделие.

Результатом решения задачи являются такие показатели, как трудоемкость и расценки по изделию на деталь (сборочную единицу) по цехам; трудоемкость и расценки единицы изделия по заводу; трудоемкость изделия по цехам, по кодам оборудования; трудоемкость и расценки изделия по видам работ; трудоемкость изделия по видам норм; трудоемкость изделия по разрядам работ.

Расчет сводной трудоемкости и заработной платы производится на все детали и сборочные единицы, предусмотренные планом собственного производства, и на те детали и сборочные единицы, ко-

которые поставляются по кооперированным поставкам и подвергаются доработкам на данном предприятии.

Получаемая в результате решения информация используется: отделом труда и заработной платы для учета, анализа и планирования снижения трудоемкости изготовления изделий по цехам; планово-экономическим отделом для расчета планов по труду по заводу (по цехам) и для определения себестоимости товарной продукции;

отделом главного технолога для расчета производственной мощности, пропускной способности и загрузки оборудования.

При решении данной задачи также формируется хранимый массив сводной трудоемкости изделия по цехам по группам оборудования для задачи «Расчет загрузки оборудования и пропускной способности». Содержание хранимого массива приведено в табл. 52.

Т а б л и ц а 52

Реквизит	Длина реквизита, знаки
Код изделия	9(11)
Вариант исполнения	9(2)
Код цеха	9(2)
Код группы оборудования	9(5)
Трудоемкость изделия с учетом норм обслуживания станков, ч	9(6), 9(2)

Исходные данные для решения задачи:

код детали (сборочной единицы) код варианта исполнения, код единицы нормирования, код цеха, код участка, код номера операции, код оборудования, код тарифной сетки, код вида норм, норма обслуживания станков, разряд работ, норма времени (мин), расценка (руб.) — из Справочника пооперационных норм времени и расценок;

код изделия (сборочной единицы), код варианта исполнения изделия, краткое наименование изделия, детали, сборочной единицы — из Справочника наименований изделий и их составных частей;

код цеха, наименование групп, код оборудования — из Справочника групп оборудования;

массив применяемости (см. § 4), полученный в результате решения задачи «Расчет применяемости»;

оперативная информация (см. § 4).

Ниже приведены формулы для решения задачи «Расчет сводных трудовых нормативов» для каждой из выходных табуляграмм.

А. Расчет сводной трудоемкости и расценок на изделие по цехам (форма № 17) производится по формулам:

а) вычисление трудоемкости на деталь (сборочную единицу)

$T_{дн}$:

СВОДНАЯ ТРУДОЕМКОСТЬ И РАСЦЕНКА НА ИЗДЕЛИЕ В РАЗРЕЗЕ ДЕТАЛЕЙ (СБОРОЧНЫХ ЕДИНИЦ) ПО ЦЕХАМ 197... г.					Наименование изделия				
					Код изделия				
					Вариант исполнения				
Код цеха	Деталь (сборочная единица)		Вариант исполнения	Количество на изделие	Трудовоемкость на 1000 изделий, мин			Расценка на 1000 изделий, руб.	
	наименование	код			на деталь	на изделие	на изделие с учетом норм обслуживания	на деталь	на изделие
1	2	3	4	5	6	7	8	9	10

Итого по цеху:

× × × × ×

Итого по заводу:

× × × × ×

$$T_{ri} = \sum_j t_{rij} \cdot l$$

где t_{rij} — норма времени на j -й операции i -й детали (сборочной единицы) по r -му цеху;

l — единица нормирования;

б) вычисление расценки на деталь (сборочную единицу) R_{ri} :

$$R_{ri} = \sum_j R_{rij} \cdot l$$

где R_{rij} — расценка на j -й операции i -й детали по r -му цеху;

в) вычисление трудоемкости на деталь (сборочную единицу) с учетом норм обслуживания станков T'_{ri} :

$$T'_{ri} = \sum_r t_{rij} \cdot m_{ij} \cdot l$$

где m_{ij} — норма обслуживания станков на j -й операции i -й детали (сборочной единицы);

г) вычисление трудоемкости на изделие T_{kr} :

$$T_{kr} = \sum_{i \in k} T_{ri} \cdot n + \sum_j t_{krj} \cdot l$$

где n — число одноименных деталей в изделии;

t_{krj} — норма времени j -й операции на сборку k -го изделия по r -му цеху;

д) вычисление расценки на изделие R_{kr} :

$$R_{kr} = \sum_{i \in k} R_{ri} \cdot n + \sum_j R_{krj} \cdot l$$

где R_{krj} — расценка j -й операции на сборку k -го изделия по r -му цеху;

е) вычисление трудоемкости на изделие с учетом норм обслуживания T'_{kr}

$$T'_{kr} = \sum_{i \in k} T'_{ri} \cdot n_i$$

Форма № 18

СВОДНАЯ ТРУДОЕМКОСТЬ И РАСЦЕНКА ЕДИНИЦЫ ИЗДЕЛИЯ ПО ЗАВОДУ В РАЗРЕЗЕ ЦЕХОВ 197__г.		Наименование изделия	
		Код изделия	
		Вариант исполнения	
Код цеха	Трудоемкость, ч	Расценка, руб.	
1	2	3	

Итого по заводу:

×

×

Б. Расчет сводной трудоемкости T_k и расценки единицы изделия R_k по заводу (форма № 18) осуществляются путем суммирования значений трудоемкости и расценки на изделие по цехам соответственно по формулам:

$$T_k = \sum_r T_{kr};$$

$$R_k = \sum_r R_{kr}.$$

Форма № 19

СВОДНАЯ ТРУДОЕМКОСТЬ ИЗДЕЛИЯ В РАЗРЕЗЕ ЦЕХОВ ПО ГРУППАМ ОБОРУДОВАНИЯ 197__г.			Наименование изделия	
			Код изделия	
			Вариант исполнения	
Код цеха	Группа оборудования		Трудоемкость с учетом нормы обслуживания станков на 1000 изделий, ч	
	Наименование	Код		
1	2	3	4	

В. Расчет сводной трудоемкости изделия по цехам и по группам оборудования (форма № 19) выполняется с учетом норм обслуживания согласно следующим формулам:

$$T'_{rqi} = \sum_j t_{rqi} \cdot m_{ij} \cdot l_j$$

где T'_{rqi} — трудоемкость с учетом нормы обслуживания i -й детали (сборочной единицы) q -й группы оборудования в r -м цехе;

t_{rqi} — норма времени на j -й операции i -й детали (сборочной единицы) по q -й группе оборудования в r -м цехе;

m_{ij} — норма обслуживания станков i -й детали на j -й операции;

$$T'_{krq} = \sum_{i \in k} T'_{rqi} \cdot n,$$

где T'_{krq} — трудоемкость по k -му изделию q -й группы оборудования в r -м цехе;

n — число одноименных деталей (сборочных единиц) в изделии.

$$T'_{rq} = \sum_k T'_{krq},$$

где T'_{rq} — трудоемкость q -й группы оборудования в r -м цехе.

Форма № 20

СВОДНАЯ ТРУДОЕМКОСТЬ И РАСЦЕНКА ИЗДЕЛИЯ ПО ЗАВОДУ В РАЗРЕЗЕ ЦЕХОВ ПО ВИДАМ РАБОТ			Наименование изделия	
			Код изделия	
			Вариант исполнения	
" " " 197__ г.				
Код цеха	Вид работ		Трудоемкость на 1000 шт., ч	Расценка на 1000 шт., руб.
	Наименование	Код		
1	2	3	4	5

Г. Расчет сводной трудоемкости и расценки изделия по видам работ (форма № 20) производится по следующим формулам:

а) расчет трудоемкости по видам работ на деталь (сборочную единицу) T_{irs} :

$$T_{irs} = \sum_j t_{irsj} \cdot l,$$

где t_{irsj} — норма времени на j -й операции i -й детали (сборочной единицы) по s -му виду работ в r -м цехе;

б) расчет расценки по видам работ на деталь (сборочную единицу) R_{irs} :

$$R_{irs} = \sum_j R_{irsj},$$

где R_{irsj} — расценка i -й детали (сборочной единицы) на j -й операции по s -му виду работ в r -м цехе;

в) расчет трудоемкости по видам работ на изделие T_{hrs} :

$$T_{hrs} = \sum_{i \in k} T_{irs} \cdot n + \sum_j t_{hrs j} \cdot l,$$

где $t_{hrs\,j}$ — норма времени на сборку k -го изделия на j -й операции по s -му виду работ в r -м цехе;

г) расчет расценки по видам работ на изделие R_{hrs} :

$$R_{hrs} = \sum_{i \in k} R_{irs} \cdot n + \sum_j R_{hrs\,j} \cdot l,$$

где $R_{hrs\,j}$ — расценка k -го изделия на j -й сборочной операции по s -му виду работ в r -м цехе;

д) расчет трудоемкости по видам работ по заводу T_{hs} :

$$T_{hs} = \sum_r T_{hrs} ;$$

е) расчет расценки по видам работ по заводу R_{hs} :

$$R_{hs} = \sum_r R_{hrs} .$$

Форма № 21

СВОДНАЯ ТРУДОЕМКОСТЬ ИЗДЕЛИЯ ПО ВИДАМ НОРМ " " " 197__ г.		Наименование изделия	
		Код изделия	
		Вариант исполнения	
Код		Трудоемкость на 1000 изделий, ч	Удельный вес вида норм, %
цеха	вида норм		
1	2		
		3	4

Д. Расчет сводной трудоемкости изделия по видам норм (форма № 21) производится по следующим формулам:

а) расчет трудоемкости по видам норм на деталь (сборочную единицу) T_{wri} :

$$T_{wri} = \sum_j t_{wrij} \cdot l,$$

где t_{wrij} — норма времени j -й операции i -й детали (сборочной единицы) в r -м цехе по w -му виду норм;

б) расчет трудоемкости по видам норм на изделие T_{hwr} :

$$T_{hwr} = \sum_{i \in k} T_{wri} \cdot n + \sum_j t_{hwrj} \cdot l,$$

где t_{hwrj} — норма времени j -й сборочной операции k -го изделия в r -м цехе по w -му виду норм;

в) расчет сводной трудоемкости по видам норм по изделию T_{hw} :

$$T_{hw} = \sum_r T_{hwr} ;$$

г) расчет удельного веса каждого вида норм в итоговой трудоемкости по заводу Y_{hw} (%):

$$Y_{hw} = \frac{T_{hw} \cdot 100}{\sum_w T_{hw}} .$$

СВОДНАЯ ТРУДОЕМКОСТЬ ИЗДЕЛИЯ ПО РАЗРЯДАМ РАБОТ 197__ г.							Наименование изделия		
							Код изделия		
							Вариант исполнения		
Код цеха	Трудовоемкость по разрядам работ, ч						Итого трудовоемкость по разрядам	Разрядочасы	Средний разряд работ
	1	2	3	4	5	6			
1	2	3	4	5	6	7	8	9	10

Итого по заводу: X X X X X X X X X

Е. Расчет сводной трудоемкости изделия по разрядам работ (форма № 22) производится по следующим формулам:

а) расчет трудоемкости по разрядам работ на деталь T_{pri} :

$$T_{pri} = \sum_j t_{pri,j} \cdot l_i$$

где $t_{pri,j}$ — норма времени j -й операции i -й детали (сборочной единицы) в r -м цехе по p -му разряду работ;

б) расчет трудоемкости по разрядам работ на изделие T_{kpr} :

$$T_{kpr} = \sum_r T_{kpr,r}$$

здесь

$$T_{kpr} = \sum_{i \in k} T_{pri} \cdot n_i + \sum_j t_{kpr,j} \cdot l_k$$

где T_{kpr} — трудоемкость k -го изделия в r -м цехе по p -му разряду работ;

$t_{kpr,j}$ — норма времени j -й операции на сборку k -го изделия в r -м цехе по p -му разряду работ.

Как видно из рис. 20, технологическая схема решения задачи реализована следующими программами: LB501, LB202, LB403, LB404, LB205, LB406, LB407, LB208, LB409, LB210, LB511 и LB412.

С помощью программы LB501 осуществляется ввод оперативной информации с перфокарт и на основании информации, находящейся в массивах PREDMET, SPECIFI, RABMEST, MARKART, формируется массив LBC51. При решении задачи по первому варианту перечень кодов рассчитываемых изделий перфорируется по схеме, представленной в табл. 48.

Программы LB202, LB205, LB208 и LB210 сортируют соответствующие массивы.

Программа LB403 производит выдачу на печать сводных трудоемкостей и расценок на изделие в разрезе деталей, сборочных

единиц по цехам (форма № 17) и формирует массив, на основании которого выдается на печать форма № 18.

Программа LB404 осуществляет выдачу на печать формы № 18, а программа LB406 — сводной трудоемкости изделия в разрезе цехов по группам оборудования (форма № 19), формирует массив LBC55 для печати формы № 20, массив LBK53, используемый при решении задачи «Расчет загрузки оборудования и пропускной способности», и массив LBK54, необходимый для решения задачи «Расчет трудоемкости производственной программы по нормам (по цехам на год, квартал, месяц)».

Программа LB407 производит выдачу на печать формы № 20.

Программа LB409 осуществляет вывод на печать формы № 21.

Программа LB511 производит расчет сводных трудовых нормативов на изделие по цехам в разрезе видов норм и формирует массивы «Сводная трудоемкость на изделие по видам норм в разрезе цехов» (массив LBC58) и «Сводная трудоемкость по заводу в разрезе цехов» (массив LBC59) для печати формы № 22.

Программа LB412 осуществляет вывод на печать формы № 22.

§ 7. Расчет нормативных калькуляций

Общее описание. Задача «Расчет нормативных калькуляций» включает расчеты нормативных калькуляций на детали (сборочные единицы) в разрезе технологических операций и на изделие в целом. На основании результатов рассматривается цеховая нормативная себестоимость деталей и сборочных единиц и полная нормативная себестоимость изделия.

Для определения указанных по предприятию нормативных затрат устанавливаются трудовые и материальные затраты на детали (сборочные единицы) в разрезе технологических операций, а также косвенные затраты на детали (сборочные единицы) в разрезе технологических операций. Нормативная калькуляция составляется на все оригинальные, стандартные и унифицированные детали, имеющиеся в справочнике состава изделия.

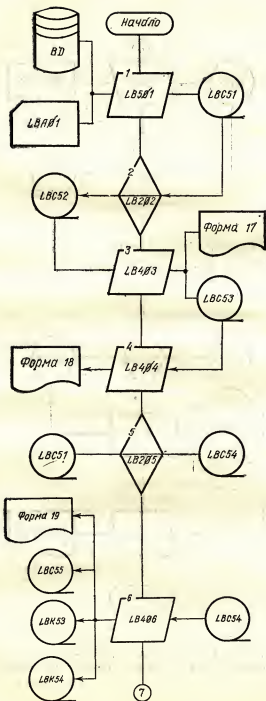
Ниже приведены табуляграммы по данной задаче (формы № 23, 24 и 25).

Исходными данными для решения задачи являются:

код единицы измерения, код единицы нормирования, чистый вес, норма расхода расчетная (чистый вес без ТНП), норма расхода с учетом отходов в заготовительном отделении, норма расхода с учетом ТНП и код цеха — из ведомости подетальных норм расхода материалов;

код детали (сборочной единицы), код варианта исполнения, код единицы нормирования, код цеха, код участка, код номера операции, код группы оборудования, код профессии, код тарифной сетки, код вида норм, норма обслуживания станков, разряд работ, норма времени (мин) и расценка (руб.) — из маршрутной карты;

код изделия, сборочной единицы, код варианта исполнения, код цеха, код участка, код номера операции, код детали, код варианта



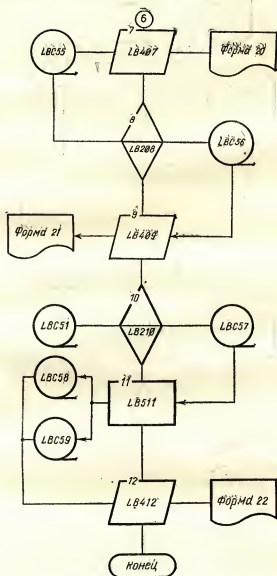


Рис. 20. Принципиальная блок-схема решения задачи «Расчет сводных трудовых нормативов»

НОРМАТИВНАЯ КАЛЬКУЛЯЦИЯ НА ДЕТАЛИ									
НА _____ КВАРТАЛ 197 ____ г.									
Код пеха	Код номера операции	Основные материалы, нарастающий итог	Возвратные отходы, нарастающий итог	Материалы за вычетом отходов, нарастающий итог	Заработная плата, нарастающий итог	Заработная плата лампового пеха, нарастающий итог	Нормативная себестоимость по прямым затратам, нарастающий итог	Цеховые расходы, нарастающий итог	Цеховая нормативная себестоимость, нарастающий итог
1	2	3	4	5	6	7	8	9	10
(7)	(8)	(13)	(12)	(13)	(12)	(13)	(13)	(13)	(13)

[illegible]

исполнения детали, количество деталей, входящих в комплект, и код цеха-поставщика — из комплектовочной карты;

краткое наименование материала, характеристика материала (ГОСТ, ТУ, марка, сорт, размер), код материала, код балансового счета, код единицы измерения, цена в рублях за 197—г., цена в рублях за 197—г., цена в рублях за 197—г., цена в рублях за 197—г. — из номенклатуры-ценника материалов;

код отхода, код единицы измерения и цена (руб.) — из ценника на отходы:

НОРМАТИВНАЯ КАЛЬКУЛЯЦИЯ НА ИЗДЕЛИЯ

НА _____ КВАРТАЛ 197 ____ г.

Основные материалы за минусом отходов	Возвратные отходы	Основные материалы	Полученные полуфабри- каты	Потери на покупные полуфабрикаты	Транспортно-загото- вительные отходы	Основная заработная плата	Цеховые расходы	Общезаводские расходы	Расходы по подготовке и освоению производ- ства	Прочие производствен- ные расходы	Итого заводская себе- стоимость	Внепроизводственные расходы	Полная себестоимость
1	2	3	4	5	6	7	8	9	10	11	12	13	14
(8)	(7)	(9)	(9)	(7)	(8)	(9)	(7)	(7)	(8)	(7)	(9)	(8)	(10)

код детали, полуфабриката, процент технологических потерь — из справочника потерь на полуфабрикаты;

код калькуляционной статьи затрат и процент косвенных от-
ходов — из ведомости косвенных расходов по калькуляционным
статьям;

код детали, код материала, цена золота, переведенная в соль,
цена отходов, коэффициент перевода золота при утере — из ценни-
ка на драгоценные металлы;

код групп изделий, код изделия, код механизма, код внешнего
оформления, код маркировки механизма и код маркировки внеш-
него оформления — из справочника состава часов;

код изделия, код варианта исполнения, код внешнего оформ-
ления, код варианта исполнения, код корпуса в сборке, код ва-
рианта исполнения, код корпусного кольца со втулкой, код вариан-
та исполнения, код головки заводной с прокладкой и шайбой, код
варианта исполнения, код циферблата в сборе, код варианта ис-
полнения, код стрелки секундной со втулкой, код варианта испол-
нения, код стрелки минутной, код варианта исполнения, код стрел-
ки часовой, код варианта исполнения, код маркировки изделия и
код варианта исполнения — из справочника варианта исполнения
часов и внешних оформлений;

код изделия, краткое наименование изделия — из справочника
наименований изделий и их составных частей;

код изделия, код варианта исполнения, код детали, сборочной
единицы (что входит), код варианта исполнения, код сборочной
единицы (куда входит), код варианта исполнения, количество на
сборочную единицу, количество на изделие, ступень диспозиции —
из задачи «Расчет применяемости»;

код детали, код варианта исполнения, код чистого металла, возвратные отходы в чистом металле и норма расхода в чистом металле — из задачи «Расчет норм расхода драгоценных металлов».

При решении данной задачи формируется хранимый массив заработной платы на операции по деталям изделия (табл. 53), используемый для задачи «Расчет ценника на брак и утерю».

Таблица 53

Реквизит	Длина реквизита, знаки
Код детали	9(11)
Код изделия (сборочной единицы)	9(11)
Код цеха	9(2)
Код номера операции	9(3)
Заработная плата на операцию	9(4), 9(2)
Заработная плата, нарастающий итог	9(4), 9(2)

Алгоритм решения задачи. Как и при описании алгоритмов решения ранее рассмотренных задач, приведем формулы для расчета данных выходных табуляграмм.

А. Расчет нормативной калькуляции на детали (форма № 23) производится по следующим формулам:

а) определение стоимости основного материала на деталь S_{imrj} :

$$S_{imrj} = N_{imrj} \cdot C_m,$$

где N_{imrj} — норма расхода m -го материала для i -й детали r -го цеха на j -й операции;

C_m — цена m -го материала.

Как правило, основной материал расходуется только на первой операции ($j=1$), поэтому далее операция не указывается;

б) определение стоимости драгоценных металлов на деталь S'_{imr} :

$$S'_{imr} = N'_{imr} \cdot C'_m,$$

где N'_{imr} — норма расхода m -го драгоценного металла на i -ю деталь r -го цеха;

C'_m — цена драгоценного металла.

Чтобы определить операцию золочения или серебрения и отнестись стоимость драгоценного металла на эту операцию, необходимо найти код оборудования и цеха в маршрутной карте, которые оговариваются при разработке постановок задач и соответствующим образом учитываются при программировании;

в) определение стоимости отходов по основному материалу O_{imr} :

$$O_{imr} = (M'_{imr} - M_{imr}) \cdot D_m,$$

где M'_{imr} — «черновой» вес m -го материала i -й детали r -го цеха;

M_{imr} — «чистый» вес m -го материала i -й детали r -го цеха;
 D_m — цена отхода m -го материала;

г) определение стоимости отходов по драгоценным металлам O'_{imr} :

$$O'_{imr} = N'_{imr} \cdot D'_m,$$

где N'_{imr} — норма возвратных отходов m -го драгоценного материала по i -й детали r -го цеха;

D'_m — цена отхода m -го драгоценного материала;

д) определение стоимости материалов за вычетом отходов S''_{imr} :

$$S''_{imr} = \begin{cases} S_{imr} - O_{imr}, \\ S'_{imr} - O'_{imr}; \end{cases}$$

ж) определение нормативной себестоимости по прямым затратам S'''_{irj} :

$$S'''_{irj} = S''_{irj} + Z_{jir},$$

где S'_{irj} — стоимость материала, использованного на j -й операции по i -й детали r -го цеха*;

Z_{jir} — заработная плата на i -ю деталь j -й операции по r -му цеху;

з) определение цеховых расходов на j -ю операцию для всех i -х деталей, обрабатываемых в r -м цехе, R_r :

$$R_r = \sum_i \sum_j \frac{P_j \cdot Z_{jir}}{100},$$

где P_j — плановый процент цеховых расходов на j -й операции;

и) определение цеховых транспортно-заготовительных расходов для всех деталей, обрабатываемых в r -м цехе, R'_r :

$$R'_r = \sum_i \sum_j \frac{S'_{irj} \cdot P'_j}{100},$$

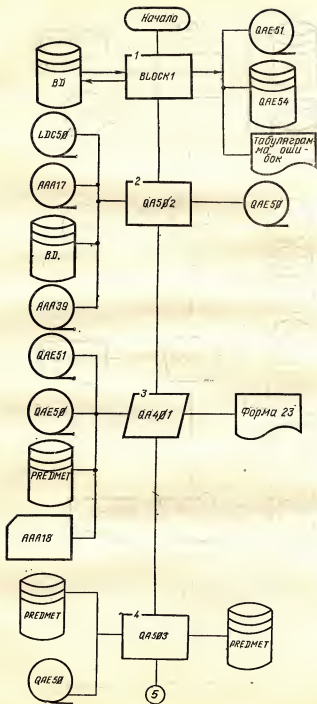
где P'_j — процент транспортно-заготовительных расходов на j -ю операцию;

к) определение цеховой нормативной себестоимости S_r^N (для всех i -х деталей, обрабатываемых в r -м цехе):

$$S_r^N = \sum_i \sum_j S''_{irj} + R_r + R'_r.$$

Б. Расчет нормативной калькуляции на сборочные единицы (форма № 24) производится по следующим формулам:

* Обозначения S'_{irj} и S'_{imr} фактически равносильны, так как это есть стоимость m -го материала по i -й детали на j -й операции по r -му цеху (S'_{imrj}).



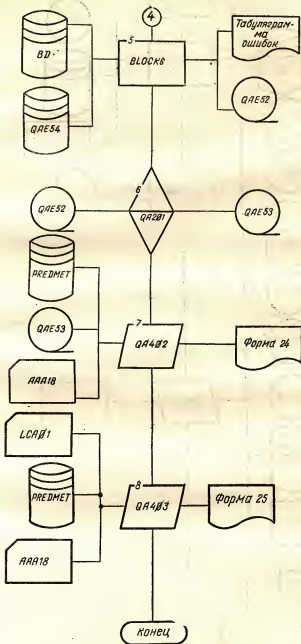


Рис. 21. Принципиальная блок-схема решения задачи «Расчет нормативных калькуляций»

а) определение предварительных затрат на основные материалы за вычетом отходов S_{hrs} :

$$S_{hrs} = \sum_{i \in h} S_{imr} \cdot n,$$

где n — применяемость деталей в сборочной единице на данной операции сборки.

Код присоединения на данной операции сборки деталей (сборочных единиц), а также их применяемость определяется по комплектовочной карте;

б) определение предварительных затрат на заработную плату с учетом применяемости деталей (сборочных единиц) на данной операции Z_{hrs} :

$$Z_{hrs} = \sum_{i \in h} Z_{ir} ,$$

где Z_{ir} — сумма заработной платы на i -ю деталь в r -м цехе;

в) определение стоимости покупных полуфабрикатов, входящих в сборочную единицу на данной операции сборки, F'_{hrs} :

$$F'_{hrs} = \sum_{i \in h} \frac{F_{ir} \cdot P}{100} ,$$

где F_{ir} — цена i -го покупного полуфабриката, используемого в r -м цехе;

P — процент технологических потерь на полуфабрикаты;

г) определение суммы заработной платы на сборочную единицу с первой по s -ю операцию сборки включительно Z'_{hrs} :

$$Z'_{hrs} = Z_{hrs} + Z_s .$$

где Z_s — заработная плата на сборочную работу на s -й операции;

д) определение нормативной себестоимости по прямым затратам N_{hrs} :

$$N_{hrs} = S_{hrs} + Z'_{hrs} + F'_{hrs} .$$

Примечания: 1. Цеховая нормативная себестоимость сборочной единицы определяется суммированием нормативной себестоимости по прямым затратам и цеховым расходам и транспортно-заготовительных расходов.

2. Нормативная калькуляция на механизм изделия, на внешнее оформление рассчитывается аналогично нормативной калькуляции на сборочную единицу.

В. Расчет нормативной калькуляции на изделие (форма № 25) производится путем суммирования прямых статей затрат (основные материалы за минусом отходов, покупные полуфабрикаты, основная заработная плата) по механизму и внешнему оформлению часов.

Назначение программ, приведенных на рис. 21, указано в табл. 54.

Таблица 54

Имя программы	Назначение программы
BLOCK1	Расчет стоимости основных материалов и заработной платы
QA502	Расчет стоимости драгоценных материалов
QA401	Печать табуляграммы нормативных калькуляций на деталь
QA503	Подключение стоимости драгоценных материалов к соответствующим сборочным единицам
QA504	Расчет нормативных калькуляций по прямым затратам на сборочную единицу
QA201	Сортировка массива
QA402	Печать табуляграммы нормативных калькуляций на сборочные единицы
QA403	Печать табуляграммы нормативных калькуляций на изделие

Таблица 55

Реквизит	Длина реквизита, знаки
Код детали	9(11)
Код цеха	9(2)
Код операции	9(3)
Код оборудования	9(5)
Наименование детали (сборочной единицы)	X(20)
Стоимость основных материалов	9(3), 9(2)
Стоимость возвратных отходов	9(2), 9(2)
Стоимость материалов по цене отходов	9(3), 9(2)
Заработная плата	9(4), 9(2)
Заработная плата данного цеха	9(4), 9(2)

Таблица 56

Реквизит	Длина, реквизита, знаки
Код сборочной единицы	9(13)
Код цеха	9(2)
Код операции	9(3)
Наименование детали (сборочной единицы)	X(20)
Стоимость материалов (отходов)	9(4), 9(3)
Стоимость материалов по цене отходов	9(4), 9(3)
Заработная плата входящих компонент	9(4), 9(3)
Стоимость полуфабрикатов	9(4), 9(3)
Стоимость сборки	9(4), 9(3)
Заработная плата данного цеха	9(4), 9(2)
Стоимость драгоценных металлов (отходы)	9(4), 9(3)

Схемы перфорации формируемых в процессе решения задачи массивов приведены ниже: массива стоимости основных материалов и заработной платы QAE51 — в табл. 55, массива нормативных калькуляций по прямым затратам на сборочную единицу QAE52 — в табл. 56, массива стоимости драгоценных материалов QAE50 — в табл. 57, массива затрат по цеху QAE54 — в табл. 58, массива калькуляционных статей AAA18 — в табл. 59.

Таблица 57

Реквизит	Длина реквизита, знаки
Код детали (сборочной единицы)	9(13)
Код цеха	9(2)
Код операции	9(3)
Стоимость драгоценных металлов	9(4), 9(3)
Стоимость отходов	9(4), 9(3)
Стоимость при браке	9(4), 9(3)
Стоимость при утере	9(4), 9(3)
Стоимость для калькуляции брака	9(4), 9(3)

Таблица 58

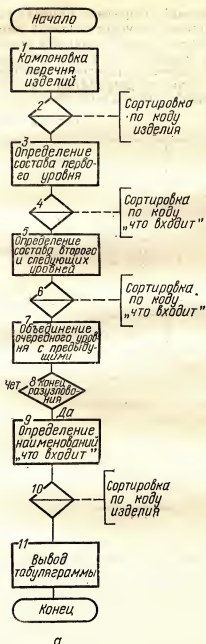
Реквизит	Длина реквизита, знаки
Код цеха	9(2)
Заработная плата цеха (общая)	9(4), 9(3)
Заработная плата цеха (по узлам)	9(4), 9(3)
Норма времени цеха (для узлов)	9(4), 9(3)

Таблица 59

Реквизит	Номера колонок на ПК	Длина реквизита, знаки
Код калькуляционной статьи	1—2	9(2)
Процент по статье	3—6	9(3), 9(2)

§ 8. Особенности проектирования информационной базы АСУП с использованием СУБД

Информационная база порождается задачами АСУП и служит основой для их решения, поэтому основным назначением банка данных является повышение эффективности разработки, внедрения и функционирования не отдельной задачи, а системы в целом. Это означает сокращение трудоемкости разработки постановок задач, их информационной увязки, составления алгоритмов и программ решения задач, оформления документации проекта и сдачи его заказчику, внедрения системы, снижение затрат машинного



времени на отладку и функционирование АСУП, уменьшение трудоемкости создания и эксплуатации самого банка данных.

При создании и ведении информационной базы АСУП перед специалистами стоят достаточно противоречивые проблемы, и оттого, какое решение будет выбрано, во многом зависит эффективность функционирования системы. Рассмотрим некоторые из них.

Использование банка данных при проектировании программного обеспечения приводит к упрощению технологической схемы решения задач. Для примера приведем принципиальную блок-схему решения известной задачи «Расчет применяемости деталей и сборочных единиц в изделии» для

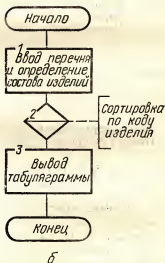


Рис. 22. Принципиальная блок-схема решения задачи «Расчет применяемости деталей и сборочных единиц в изделии»: а — при традиционном подходе; б — с использованием банка данных

ЭВМ второго* и третьего поколения (рис. 22). Из сравнения этих двух схем видно, что во втором случае количество программ сократилось более, чем в 3 раза, особенно программ сортировок. Так, если в первом случае их число равно $3+n$ (где n — число уровней вхождения деталей и узлов), то во втором случае присутствует только одна сортировка. Хотя уменьшение количества программ и не всегда ведет к сокращению времени решения задач, но в данном случае благодаря наличию банка данных время счета при условии одинакового быстродействия ЭВМ сократилось примерно в четыре раза (имеется в виду только время счета без выдачи результатов на печать). Упрощение технологической схемы значительно облегчает управление процессом решения задачи, что становится немаловажным фактором, когда число задач АСУП исчисляется несколькими десятками.

При применении банков данных поддерживается более высокий уровень достоверности данных, поэтому качество полученных результатов в процессе решения задачи значительно повышается, так как сама организация базы данных обеспечивает более высокий уровень и контроль исходной информации за счет проверки взаимосвязи между данными различных документов.

На выверку базы данных расходуется значительно меньше времени: при использовании независимых массивов данный процесс фактически заканчивается по завершении решения всех или большинства задач. В отдельных случаях такой путь остается основным критерием оценки качества базы данных, хотя он и связан с большими потерями машинного времени. Это объясняется сложностью системы контроля исходной информации при ее больших объемах и многообразии взаимосвязей. Система управления базами данных (СУБД) частично решает проблему контроля предусмотренных разработчиком взаимосвязей и в случае их отсутствия информирует об этом потребителя, чем ускоряет процесс выверки базы данных. Но СУБД не может определить всевозможные ошибки при формировании массивов. Например, она не может установить конкретные детали, не вошедшие в массив предметов, но указать число деталей, не подвергавшихся обработке, если известно их общее количество, для нее не составляет труда.

С организацией информационной базы данных в виде банка данных неразрывно связано понятие «отсутствие избыточной информации». Однако здесь речь идет об отсутствии избыточной информации, отражающей процессы и явления управляемого объекта. Так, коды изделия, детали, материала, цеха, оборудования встречаются в базе данных один раз, но это не означает, что общий объем информации базы данных будет минимальным. Наоборот, общий объем информации имеет тенденцию к увеличению из-за наличия прямых и обратных адресных ссылок, которые отражают объективно существующие зависимости между реквизитами

* Принципиальная блок-схема указанной задачи рассматривается на примере одного из часовых заводов для ЭВМ «Минск-32».

(полями) массивов банка данных. Например, между деталью и материалом, деталью и цехом, деталью и оборудованием и т. п. Наличие одного сочетания в исходных данных, например «деталь — цех», отражает два факта: в каком цехе обрабатывается деталь и какая деталь (детали) обрабатывается в цехе. В банке данных для этих целей должны существовать фактически две адресные ссылки: для детали задается адрес, где указан цех, обрабатывающий деталь (прямая адресная связь), а для цеха задается адрес, где указана деталь, обрабатываемая в данном цехе (обратная адресная связь). При наличии нескольких деталей (цехов) заполняются дополнительные поля, где указывается адрес следующей детали (цеха), относящиеся соответственно к рассматриваемому цеху (детали). Так, для банка данных АСУ Первого московского часового завода (АСУ-1-МЧЗ) можно привести следующие данные. Общий объем банка данных составляет более ста тысяч записей. Вся информация распределена между массивами банка данных следующим образом, тыс. записей: массив предметов — 16; массив спецификаций — 15; массив технологических маршрутов — 70, массив рабочих мест — 3. Коэффициент увеличения общего объема информации равен 1,5.

Как отмечалось ранее, организация конструкторско-технологической информации в виде банка данных позволяет в значительной мере уменьшить время решения задач по сравнению со случаем, когда эта же информация организована в виде независимых последовательных массивов (табл. 60).

Таблица 60

Задача	Число условных изделий*	Время решения задачи**	
		с использованием банка данных	без использования банка данных
Расчет применяемости деталей (сборочных единиц) в изделии	100	1 ч 15 мин	4 ч 20 мин
Расчет применяемости детали (сборочной единицы) в изделиях	250	3 ч 30 мин	10 ч 30 мин
Расчет материальных нормативов	100	2 ч 30 мин	3 ч 50 мин
Расчет трудовых нормативов	100	3 ч	5 ч 10 мин
Расчет нормативных калькуляций на детали, сборочные единицы, изделия по цехам	250	5 ч	19 ч *
Расчет межцеховых маршрутов	250	2 ч	2 ч 30 мин

* Условное изделие характеризуется следующими показателями:
среднее число составных частей (деталей и узлов) в одном изделии — 160;
среднее число операций на обработку одной детали — 20;
среднее число цехов по технологическому маршруту — 6.

** Время решения задач включает и время вывода табуляграмм.

При проектировании АСУП с банком данных значительно сокращаются возможности разрабатывать программное обеспечение АСУП параллельно, поскольку отладку программ можно эффек-

тивно вести при наличии массивов банка данных, организованных на машинных носителях. Это значит, что процесс рабочего проектирования имеет тенденцию к удлинению. Кроме того, в случае внесения изменений в структуру банка данных корректировке подвергаются все задачи, использующие банк данных, что также ведет к увеличению сроков разработки АСУП. Избежать в максимальной степени указанных потерь времени, материальных и трудовых затрат можно при условии разработки высококачественного технического проекта как по нормативному хозяйству, так и по функциональным подсистемам. Тем более что документация технического проекта в АСУП фактически является рабочей проектной документацией. Однако многие специалисты недооценивают важности технического проекта, в связи с чем возникают неточности в нем.

Для повышения качества технического и рабочего проектов в первую очередь необходима стандартизация форм нормативно-справочной документации (НСД). Требуется выбрать набор НСД или несколько его версий для определенного круга родственных предприятий с одинаковым типом производства по той или иной отрасли. Это не исключает введения единых форм НСД для всех предприятий страны, но только в тех случаях, где это необходимо и не усложняет проблему подготовки данных и организацию управления.

На этапе функционирования системы много машинного времени расходуется на функции ведения базы данных (БД): корректировку, периодическую запись и перезагрузку БД или ее восстановление. Хранение данных в БД в случае их большого объема требует большого количества дисководов (или более емких пакетов дисков), а реализация функции СУБД — достаточно большого объема основной памяти ЭВМ.

Общепризнанного мнения относительно степени централизации информации еще не выработано. Теоретические расчеты и имеющийся у нас в стране и за рубежом опыт показывают, что основное влияние на степень централизации оказывают период стабильности, регламент и объем поступления данных из объекта управления и использования при решении задач информации из БД. Интеграция в одну базу разных по характеру данных неотвратно снижает эффективность выполнения ряда процедур обработки данных, резко повышает стоимость поддержания такой базы. В связи с этим наиболее предпочтительным представляется вариант организации информационной базы в АСУП в виде нескольких локальных баз в зависимости от частоты поступления и использования данных и их объема. Каждая база данных может быть организована своими методами и даже обрабатываться своими СУБД. Для обеспечения целостности информационного процесса должен быть организован обмен информацией между базами. Опыт построения АСУП показывает, что с точки зрения регламента использования объема данных рекомендуется создавать четыре группы баз данных:

БД1, содержащая информацию перспективного (год и более) планирования и учета. В этой группе накапливаются статистические данные за большой период времени (год и более), а также другая информация с небольшим количеством сеансов обращения к ней или изменений (включая пополнение) в течение года;

БД2, объединяющая информацию для текущего (от месяца до года) планирования и учета;

БД3, включающая информацию для оперативного планирования, учета и регулирования;

БД4, содержащая информацию для функционирования системы по управлению технологическими процессами.

Каждая группа может быть разбита на несколько информационно независимых или слабо зависимых баз, например по подсистемам. Это упростит их создание и ведение.

Задачи, использующие информацию БД1, решаются сравнительно редко. Объем данных этой БД достаточно большой с длительным временем хранения, поэтому такую базу данных наиболее целесообразно организовать в виде независимых массивов с последовательным методом доступа. Магнитные ленты и диски с указанной информацией целесообразно хранить в архиве и подключать к ЭВМ к моменту решения задачи или изменения БД1.

Вся нормативно-справочная информация и основной объем промежуточных данных, создаваемых при функционировании АСУП, включаются в БД2. Поскольку отдельные подсистемы слабо информационно связаны с другими подсистемами, то для них целесообразно разрабатывать локальные базы данных, входящие в группу БД2. Например, база данных по подсистеме реализации и сбыту продукции имеет почти независимую нормативно-справочную информацию и только отдельные промежуточные результаты используются в других подсистемах. Такие общие для нескольких подсистем данные, как код готовой продукции, ее наименование и прочая информация, можно продублировать в нескольких массивах. Из-за малого их объема время ведения баз данных практически не изменяется, а продолжительность функционирования задач значительно сокращается.

Опыт разработки АСУП показал, что для организации и ведения БД2 достаточно хорошо приспособлены системы БАСТАИ и САВИ.

БД3 ориентирована на работу в реальном масштабе времени и должна периодически перезагружаться на основании БД2 с сохранением отдельных собственных данных. Эта база по сравнению с БД2 включает значительно меньший объем данных, но ее построение должно обеспечивать поиск необходимой информации, поэтому в БД3 включается только та информация, которая необходима для процедур оперативного характера. Система управления базой данных, эффективная для управления БД1 и БД2, может оказаться непригодной для БД3, так как последняя должна удовлетворять следующим условиям:

а) данные, обрабатываемые в системе реального времени, должны быть централизованы в одной базе;

б) для обеспечения быстрой реакции на запрос допускается хранение избыточной системной (для обеспечения адресных ссылок) и производственной информации;

в) необходима надежная система защиты и восстановления данных;

г) система должна обеспечивать корректировку данных в базе в соответствии с темпом поступления в нее информации.

Отсутствие формализованных методов создания баз данных, информационно связанных с решением большого спектра задач АСУП, требует выделения на их разработку группы высококвалифицированных специалистов. Их основная задача — определить состав и структуру баз данных, методы их защиты, загрузки и ведения, а также выбрать СУБД. Такие специалисты должны хорошо представлять алгоритм и режим решения задач, включенных в АСУП, знать возможности конструирования и создания базы данных и принципы работы управляющей программы СУБД.

Исходя из опыта разработки АСУП и предполагая, что специалисты достаточно знакомы с системой программирования ЕС ЭВМ и хорошо владеют выбранной ими СУБД, можно указать следующие работы по проектированию информационной базы:

изучение состава и содержания документов фонда НСИ;

уточнение содержания отдельных задач или алгоритмов их решения;

определение состава и структуры массивов банка данных и их взаимосвязи с исходными нормативно-справочными документами;

разработка схем (макетов) перфорации;

выбор технологической схемы обработки и методов контроля информации;

составление комплексного тестового примера, обеспечивающего отладку программ подготовки загрузки и изменений банка данных и программ решения задач;

разработка программ подготовки загрузки и изменений банка данных (или соответствующих таблиц описания данных);

подготовка исходных данных тестового примера на первичных машинных носителях;

запись исходной информации тестового примера на МЛ и контроль;

генерирование операционной системы банка данных (генерирование СУБД);

отладка программ подготовки, загрузки и изменений банка данных;

загрузка банка данных (тестовым примером);

анализ результатов загрузки и корректировка;

своевременное обеспечение проектировщиков задач информацией о состоянии базы данных и ее изменениях.

Подготовка информации, разработка и отладка программ задач ведутся параллельно с вышеперечисленными работами о

учетом полученных результатов тестирования программ организации и ведения банка данных. При получении положительных результатов анализа и при наличии подготовленной на машинных носителях информации можно приступать к организации банка данных.

Оценка состояния банка данных проводится на основании контрольных листингов, предусматриваемых при работе программ организации банка данных и его изменений, по распечаткам массивов банка данных и, наконец, по результатам отладки и решения задач.

На стадии функционирования АСУП появляется ряд дополнительных функций, выполняемых специалистами службы баз данных:

- актуализация базы данных, включая ее восстановление и реорганизацию;

- изменение структуры базы данных согласно изменившимся или вновь возникшим потребностям;

- выполнение работ по обмену информацией между базами данных (если это предусмотрено схемой ведения баз данных).

В заключение следует отметить, что хотя применение банков данных приводит к увеличению трудовых и временных затрат на этапе внедрения АСУП, но в дальнейшем уровень функционирования системы повышается за счет сокращения времени счета задач и увеличения достоверности получаемых результатов.

ПРИЛОЖЕНИЯ

1. СТРУКТУРА СИСТЕМНОЙ ЧАСТИ ЗАПИСЕЙ МАССИВОВ БАНКА ДАННЫХ

Таблица 1

Структура системной части записей массива предметов

Номер поля	Имя поля	Название поля	Поле резервируется, если задан параметр	Длина поля, байт
1	GD#UEKAD	Адрес записи следующего предмета в порядке сортировки	—	4
2	GD#GIDNR	Код предмета	—	GDLGIDNR* 4
3	GD#ADEPS	Адрес первой записи спецификации	—	
4	GD#POZAE	Счетчик записей спецификаций	SATZZAEH	2
5	GD#ADEGS	Адрес первой записи спецификации в цепи использования предмета (цели применения)	—	4
6	GD#GKZAE	Счетчик записей в цепи применения	SATZZAEH	2
7	GD#DISPO	Степень диспозиции (упаковочное число)	DISPSTUF	2
8	GD#ADNAK	Адрес записи следующего предмета в рабочей цепи	DISPSTUF	4
9	GD#AKPRU	Контрольное поле записи предмета в рабочей цепи	DISPSTUF	LGOPRUEF
10	GD#LAUFN	Текущий номер	—	
11	GD#ADEAS	Адрес записи первой рабочей операции	VORAGKET	4
12	GD#ADLAS	Адрес записи последней рабочей операции	VORAGKET RCKPRKET	4
13	GD#AGZAE	Счетчик записей рабочих операций	VORAGKET SATZZAEH	2

* Длина поля задается параметром.

Таблица 2

Структура системной части записей массива спецификаций

Номер поля	Имя поля	Название поля	Поле резервируется, если задан параметр	Длина поля, байт
1	2	3	4	5
1	SD#ADUBS	Адрес записи компоненты в массиве предметов	—	4
2	SD#PRUBS	Контрольное поле записи компоненты	—	LGUPRUG
3	SD#ADNPS	Адрес следующей записи спецификации	—	4
4	SD#ADOBS	Адрес записи сборочной единицы в массиве предметов	—	4
5	SD#PROBS	Контрольное поле записи сборочной единицы	—	LGOPRUEF
6	SD#ADNGS	Адрес следующей записи спецификации в цепи применения предмета	—	4
7	SD#ADVGS	Адрес предыдущей записи спецификации в цепи применения предмета	RCKSEKET	4

Таблица 3

Структура системной части записей массива рабочих мест

Номер поля	Имя поля	Название поля	Поле резервируется, если задан параметр	Длина поля, байт
1	2	3	4	5
1	MD#UEKAD	Адрес записи следующего рабочего места в порядке сортировки	—	4
2	MD#MIDNR	Код рабочего места	—	MDLGIDNR
3	MD#ADEMS	Адрес первой записи рабочих операций в цепи использования рабочего места	VORSEKET	4
4	MD#MVZAE	Счетчик записей в цепи использования рабочего места	VORSEKET SATZZAEH	2

Структура системной части записей массива технологических маршрутов

Номер поля	Имя поля	Название поля	Поле резервируется, если задан параметр	Длина поля, байт
1	2	3	4	5
1	AD#ADOBS	Адрес записи предмета в массиве предметов	—	4
2	AD#PROBS	Контрольное поле записи предмета	—	LGOPRUEF
3	AD#ABGNR	Номер рабочей операции	—	ADLGAGNR
4	AD#ADNAS	Адрес записи следующей рабочей операции	—	4
5	AD#ADVAS	Адрес записи предыдущей рабочей операции	RCKPRKET	4
6	AD#PRUBS	Контрольное поле записи рабочего места	MDDATNAM	LGUPRUFM
7	AD#ADUBS	Адрес записи рабочего места в массиве рабочих мест	—	4
8	AD#ADNMS	Адрес записи следующей рабочей операции в цепи использования рабочего места	MDDATNAM VORSEKET	4
9	AD#ADVMS	Адрес записи предыдущей рабочей операции в цепи использования рабочего места	MDDATNAM VORSEKET RCKAPVWK	4

2. ПАРАМЕТРЫ ДЛЯ ГЕНЕРАЦИИ МОДУЛЕЙ ПРОГРАММ СОЗДАНИЯ И ОБСЛУЖИВАНИЯ БАНКА ДАННЫХ

В таблицах приняты обозначения:

- X — имя, которое может состоять не более чем из восьми буквенно-цифровых символов и должно начинаться с буквы;
 d — длина операнда в байтах;
 n — количество, целое число;
 a — относительный адрес в записи, первый байт в записи имеет относительный адрес 0;
 c — значение признака функции, оно может состоять из букв и цифр, максимальная длина 8 байт.

Параметры для генерации модуля LEITPROG

Параметр	Операнд	Пояснения
PROGRART PROGNAME	LEITPROG X	Указывается тип модуля Указывается имя сгенерированного модуля в библиотеке объектных модулей (генерируется карта CATALR X)
MBEINGAB	X	Указывается имя массива при вводе с МЛ (его максимальная длина семь знаков). Если параметр опущен, предполагается ввод с ПК
KENNSATZ	—	Массив на МЛ имеет стандартные метки.
BLLAENGE	d(1500)	Для массива без меток параметр опускается Указывается максимальная длина блока на МЛ. Если значение длины не задано, оно принимается равным 1500 байт
MAXSALAE	d(240)	Указывается максимальная длина записи на МЛ
LKLESERI	—	Указывается имя массива при вводе с ПК.
DFEBERGD	—	Его максимальная длина семь знаков Вслед за этим параметром нужно описать структуру входной записи на МЛ для массива предметов (МП)*
DEFINEND DFEBERSD DEFINEND	— — — }	Конец описания структуры записи Структура входной записи для МС
DFEBERMD DEFINEND	— — }	Структура входной записи для РМ
DFEBERAD DEFINEND GDDATNAM	— — X }	Структура входной записи для ТМ
GDLGIDNR	d	Указывается имя МП, состоящее из семи знаков
DISPSTUF	—	Указывается длина поля кода предмета. Если код состоит из нескольких подряд расположенных реквизитов, то задается их суммарная длина
LGOPRUEF	d	В записях МП отводится место для ступеней диспозиции (см. приложение 1, табл. 1) Указывается длина контрольного поля в записи МС для занесения сборочной единицы и в записи ТМ для занесения предмета. Если задан параметр DISPSTUF, то он задает длину контрольного поля записи предмета в рабочей цепи
VORAGKET	—	При задании параметра в записях МП будет отведено место для адреса записи первой рабочей операции. Он задается, если имеется массив ТМ
RCKPRKET	—	При задании параметра в записях МП будет отведено место для адреса записи последней рабочей операции. Параметр можно опускать

* Описание структуры записей выполняется на языке Ассемблера с помощью инструкций DS, DC, ORG.

Параметр	Операнд	Пояснения
DFSATZGD	—	Вслед за этим параметром следует описать структуру на диске части пользователя записи МП на языке Ассемблера
DEFINEND	—	Параметр указывает конец описания структуры записи МП
RCKSEKET	—	При задании параметра в записи МС будет отведено поле SD#ADVGS (см. приложение 1, табл. 2), т. е. поле адреса предыдущей записи спецификации в цепи применения предмета
LGUPRUF	d	Параметр задает длину контрольного поля для записи компоненты в записи МС
SDDATNAM	X	Указывается имя МС, состоящее из семи знаков
DFSATZSD	—	Вслед за этим параметром следует описать на языке Ассемблера структуру части пользователя на диске в записях МС
DEFINEND	—	Параметром указывается конец описания структуры записей МС
MDDATNAM	X	Указывается имя РМ, состоящее из семи знаков
MDLGIDNR	d	Параметром указывается длина кода рабочего места
VORSEKET	—	При задании параметра в записях РМ и ТМ будут отведены адресные поля для цепи использования рабочего места (см. приложение 1, табл. 3 и 4)
DFSATZMD	—	Вслед за этим параметром нужно описать структуру части пользователя в записях РМ на языке Ассемблера
DEFINEND	—	Параметром указывается конец описания структуры записей РМ
ADDATNAM	X	Указывается имя ТМ, состоящее из семи знаков
ADLGAGNR	d	Параметром задается длина номера рабочей операции
LGUPRUFM	d	Параметром задается длина контрольного поля для записи рабочего места в записи ТМ
RCKAPVWK	—	При задании параметра в записях ТМ будет отведено адресное поле для обратной цепи использования рабочего места (см. приложение 1, табл. 4)
DFSATZAD	—	Вслед за этим параметром нужно описать структуру части пользователя в записях ТМ на языке Ассемблера
DEFINEND	—	Параметром указывается .конец описания структуры записей ТМ
GDVARSLG	—	Параметр задается в том случае, если входная запись для МП состоит из переменного числа карт (при вводе с ПК) или записей на МЛ. На каждой карте или в каждой записи на МЛ должен быть указан код предмета и признак функции

Параметр	Операнд	Пояснения
GDKONSLG	n	Параметр задается в том случае, если входная запись для МП состоит из постоянного числа ПК или записей на МЛ. Если $n=1$, то вместо n надо записать STD. Код предмета и признак функции необходимо указывать только в первой записи
FKZEICHN	X	Параметром задается во входной записи имя поля, содержащего признак функции. Признак функции во всех записях должен иметь одинаковую длину (до восьми байт) и занимать одно и то же место
CDIDNTNR	X	Параметром задается имя поля во входной записи для МП, содержащего код предмета
GDFKLADN	c	Параметром задается значение признака загрузки для МП. Признак может содержать от одного до восьми алфавитно-цифровых символов, например P1. Если параметр опущен, модуль не выполняет этой функции
GDFKHNZU	c	То же для функции добавления
GDFKBERI	c	То же для функции корректировки
GDFKLSCH	c	То же для функции стирания
GDPROTOK	d	Параметром задается длина поля для помещения распакованного кода предмета (см. гл. 1, § 7, описание СПП112)
SDLGSATZ	d	Параметром задается длина записи MC на диске, равная суммарной длине системной части и части пользователя
SDLGSORT	d	Параметром задается суммарная длина признаков сортировки входных записей для MC. Признаки сортировки выбираются таким образом, чтобы упорядочить записи внутри группы записей, относящихся к одной сборочной единице. Поэтому код сборочной единицы не следует выбирать в качестве признака сортировки. Такими признаками могут служить, например, код компоненты, код цеха, номер операции и т. д. Допускается не более трех признаков сортировки. Все признаки должны быть расположены в первой физической записи, если входная запись MC состоит из нескольких ПК или записей на МЛ.
SDSORTF1	X	Параметр не требуется, если признаки расположены в записи непрерывно
SDSORTF2	X	Параметром задается во входной записи имя поля, содержащего первый признак сортировки
SDSORTF3	X	Параметром задается имя поля во входной записи для второго признака сортировки
SDKONSLG	n или STD	Параметром задается имя поля во входной записи для третьего признака сортировки
		Параметром задается количество ПК или записей на МЛ для одной входной записи MC. Код сборочной единицы и признаки сортировки должны содержаться в первой записи. Если $n=1$, задается STD

Параметр	Операнд	Пояснения
SDVARSLG	—	Параметр задается, если входная запись для МС состоит из переменного числа карт или записей на МЛ. Признак функции, код сборочной единицы и признаки сортировки должны содержаться в каждой входной записи
SDIDNRGP	X	Параметром задается во входной записи имя поля, содержащее код сборочной единицы
SDIDNRGK	X	Параметром задается во входной записи имя поля, содержащее код компоненты
SDFKLADN	c	Параметром задается значение признака функции загрузки для МС. Если параметр опущен, модуль не сможет выполнить этой функции
SDFKHNZU	c	То же для функции добавления
SDFKBERI	c	То же для функции корректировки
SDFKELOE	c	То же для функции «стирание простое»
SDFKMLOE	c	То же для функции «стирание групповое»
SDFKERSE	c	То же для функции замены. Возможна замена для кода компоненты и признаков сортировки. При этом старая и новая длины заменяемых полей должны совпадать
SDIDERGD	X	Параметром задается во входной записи МС имя поля, содержащее код новой компоненты
SDESRTFI	X	Параметром задается во входной записи МС имя поля, содержащее новый признак сортировки
SDESRTF2	X	То же для второго признака сортировки
SDESRTF3	X	То же для третьего признака сортировки
MDVARSLG	—	Параметр задается в том случае, если входные записи для РМ состоят из переменного числа ПК или записей на МЛ. При этом признак функции и код рабочего места должны содержаться в каждой записи
MDKONSLG	p или STD	Параметр задается в том случае, если входная запись для РМ состоит из постоянного числа ПК или записей на МЛ. Если p=1, вместо p надо записать STD. Признак функции и код рабочего места требуются в первой записи
MDIDNTNR	X	Параметром указывается во входной записи для РМ имя поля, содержащее код рабочего места. Код может состоять из нескольких реквизитов, например код оборудования, инвентарный номер или пех, участок и т. д.
MDFKLADN	c	Параметром задается значение признака функции загрузки для РМ. Если параметр опущен, модуль не выполняет этой функции
MDFKHNZU	c	То же для функции добавления
MDFKBERI	c	То же для функции корректировки
MDFKLSCH	c	То же для функции стирания
MDFKTREN	c	То же для функции отделения, при этом для модуля DATEIORG нужно задать параметр TREVDUBD

Параметр	Операнд	Пояснения
MDFKVERB	c	То же для функции соединения, при этом для модуля DATEIORG нужно задать параметр VERVDUBD
MDPROTOK	d	Параметром задается длина поля для помещения упакованного кода рабочего места (см. гл. 1, § 8, описание СПП23)
ADLGSATZ	d	Параметром задается длина записи ТМ на диске, равная суммарной длине системной части и части пользователя
ADLGSORT	d	Параметром задается суммарная длина признаков сортировки входных записей для ТМ. Допускается не более трех признаков сортировки. Каждый признак может состоять из нескольких полей, однако эти поля во входной записи должны быть расположены рядом. В качестве первого признака рекомендуется выбирать номер операции. Другими признаками могут быть код рабочего места, код цеха и т. д. Если входная запись состоит из нескольких ПК или записей на МЛ, то все признаки должны находиться в первой записи при постоянном числе записей и в каждой записи при переменном. Параметр не требуется, если все признаки расположены в записи непрерывно
ADSORTF1	X	Параметром задается во входной записи имя поля, содержащее первый признак сортировки
ADSORTF2	X	Параметром задается имя поля во входной записи для второго признака сортировки
ADSORTF3	X	Параметром задается имя поля во входной записи для третьего признака сортировки
ADKONSLG	p или STD	Параметр задается в том случае, если входные записи для ТМ состоят из постоянного числа ПК или записей на МЛ. Признак функции, код предмета и признаки сортировки должны находиться в первой записи. При p=1 вместо p надо задавать STD
ADVARS LG	—	Параметр задается в том случае, если входная запись для ТМ состоит из переменного числа карт или записей на МЛ. Признак функции и признаки сортировки должны присутствовать во всех записях
ADIDNRGD	X	Параметром указывается во входной записи ТМ имя поля, содержащее код предмета
ADIDNRMD	X	Параметром указывается во входной записи ТМ имя поля, содержащее код рабочего места
ADIDERMD	X	Параметром указывается во входной записи ТМ имя поля, содержащее код нового рабочего места для функции замены
ADFKLADN	c	Параметром задается значение признака функции загрузки для ТМ. Если параметр опущен, модуль не выполняет этой функции

Параметр	Операнда	Пояснения
ADFKHNZU	c	То же для функции добавления
ADFKBERI	c	То же для функции корректировки
ADFKELOE	c	То же для функции «стирание простое»
ADFKMLOE	c	То же для функции «стирание групповое»
ADFKERSE	c	То же для функции замены. Допускается замена кода рабочего места и значений признаков сортировки
ADESRTF1	X	Параметром задается во входной записи ТМ имя поля, содержащее новое значение первого признака сортировки для замены
ADESRTF2	X	То же для второго признака
ADESRTF3	X	То же для третьего признака
ZATZZAEH	—	При задании параметра для каждой цепи резервируются в базовой записи поля для счетчиков записей цепи (см. приложение 1, табл. 1—4)
MAKROAUF	—	При задании параметра в распечатке LEITPROG печатаются все макрорасширения. Если параметр опущен, печатаются только определения областей, констант и СПП. Чтобы не печатались макрорасширения в СПП, после параметра DFKONSTF нужно поставить PRINT NOGEN
ANSAROnn	—	Параметр указывает на необходимость использования СПП с номером nn. СПП помещается вслед за этим параметром, nn принимает значения от 02 до 25
ENSAROnn	—	Параметр указывает на конец СПП
DFKONSTF	—	После записи этого параметра нужно записать все константы и поля, используемые в СПП. Здесь же можно поместить программу печати контрольных таблиц загрузки и изменения массивов, запрограммированную с использованием макрокоманды AD#RU
DEFINEND	—	Параметр определяет конец перечисления констант и полей, записанных после параметра DFKONSTF

Таблица 2

Параметры генерации модуля DATEIORG для загрузки базовых массивов

Параметр	Операнда	Пояснения
PROGRART PROGNAME	DATEIORG X	Параметром задается тип модуля Параметром задается имя сгенерированного модуля в библиотеке объектных модулей (генерируется карта CATALR X)

Параметр	Операнд	Пояснения
BDALADEN	—	Параметр указывает, что требуется создание модуля для загрузки базового массива. Параметры функций изменения для этого модуля указывать нельзя
DATEINAM	X	Параметром указывается имя массива, содержащее семь знаков
LNGEIDNR	d	Параметром задается длина кода предмета или рабочего места
LNGESATZ	d	Параметром задается длина записи, включающая системную часть и часть пользователя
BLOCKFAK	n	Параметром задается количество записей в блоке для базового массива
BDSPURZY	n	Параметром задается число дорожек на цилиндре для основной области данных. При $n=9$ резервируется одна дорожка переполнения, при $n=10$ дорожки переполнения не резервируются. Добавляемые записи в последнем случае будут помещаться в область, оставшуюся свободной после загрузки массива
RAKENNZF	a	Параметром указывается относительный адрес поля отметок для стирания в части пользователя базовой записи. Длина поля равна двум байтам
RALAUFNR	a	Параметром указывается в системной части записи МП относительный адрес поля, в которое помещается текущий номер использования массива
MAKROAUF	—	При задании параметра распечатка Ассемблера будет выдаваться с макрорасширениями

Таблица 3

**Параметры генерации модуля DATEIORG
для загрузки массивов связей**

Параметр	Операнд	Пояснения
PROGRART PROGNAME	DATEIORG X	Параметром указывается тип модуля Параметром задается имя сгенерированного модуля в библиотеке объектных модулей (перфорируется карта CATALR X)
VDALADEN	—	Параметр указывает, что требуется создание модуля для загрузки массива связей. Параметры функций изменения для этого модуля указывать нельзя
DISPSTUF	—	При задании параметра предусматривается задание ступеней диспозиции для МС
RCKPRKET	—	Параметр указывает на наличие обратной цепи рабочих операций в массиве ТМ

Параметр	Операнд	Пояснения
SATZZAEH	—	При задании параметра в системных частях записей базовых массивов резервируется поле счетчика записей в цепи. Подсчет записей производится при загрузке массива связей
KEINEUBD	—	Параметр указывает, что подчиненный базовый массив отсутствует
EINUNTB	—	Параметр указывает, что для массива связей имеется подчиненный базовый массив
VORSEKET	—	Параметр указывает на необходимость создания прямой цепи использования предмета или рабочих мест
RCKSEKET	—	Параметр указывает на необходимость создания обратной цепи использования предметов или рабочих мест
UNTBDERF	—	Параметр нужно задавать, если признак сортировки записей связей находится в подчиненном базовом массиве (для МС параметр обязателен)
UEBUNTGL	—	Параметр указывает, что подчиненный и старший базовый массивы являются одним и тем же массивом. Параметр можно задавать только для МС
CDDATNAM	X	Параметром задается имя МП, состоящее из семи знаков
LAUFNUMR	—	Параметр вызывает резервирование в записях МП поля для текущего номера использования массива и должен задаваться в обязательном порядке
DEFVEDAT	—	Параметр указывает начало параметров, относящихся к массиву связей
DATEINAM	X	Параметром задается имя массива связей, состоящее из семи знаков
LGIDNRGD	d	Параметром задается длина кода предмета в записях МП
RALPRKET	d	Параметр указывает адрес поля GD#ADLAS в записи МП (см. приложение I, табл. I)
RAVPRKET	a	Параметр указывает адрес поля AD#ADVAS в записи ТМ
RAEPRKET	a	Параметр указывает адрес поля GD#ADEPS в записи МП для связи с МС или адрес поля MD#ADEAS в записи МП для связи с ТМ
RASATZAE	a	Параметр указывает адрес поля счетчика в записи МП (GD#POZAE для МС или GD#ADZAE для ТМ)
RAOLAUFN	a	Параметр указывает адрес поля в записи МП для текущего номера использования массива
RAPRUFOB	a	Параметр указывает адрес части поля GD#GIDNR, которая должна выбираться из кода предмета для сравнения с контрольным полем в записях массивов связей
LGOPRUEF	d	Параметр задает длину контрольного поля в записи связи для старшей базовой записи
RAPRUFOV	a	Параметр указывает адрес контрольного поля в записях связей (SD#PROBS для МС или AD#PROBS для ТМ)

Параметр	Операнд	Пояснения
RAOBASAT	a	Параметр указывает адрес поля в записи связи для адреса старшей базовой записи (SD#ADOBS для MC или AD#ADOBS для TM)
RANPRKET	a	Параметр указывает адрес поля для адреса следующей записи связи, относящейся к этой же старшей базовой записи (SD#ADNPS для MC или AD#ADNAS для TM)
RADSTUFE	a	Параметр указывает адрес поля в записи МП для ступеней диспозиции (GD#DISPO)
RAARBKET	a	Параметр указывает адрес поля в записи МП для адреса следующей записи в рабочей цепи (GD#ADNAK)
RAARKEPR	a	Параметр указывает адрес контрольного поля для записи предмета в рабочей цепи (SD#AKPRU)
LNGESATZ	d	Параметром задается длина записи в массиве связей
BLOCKFAK	n	Параметром задается количество записей в блоке для массива связей
ANZUNTB	n	Параметром задается количество подчиненных базовых массивов. Для MC $n=1$, для TM $n=1$ или 0, если PM отсутствует
RANSKETI	a	Параметр указывает адрес поля в записи связи для адреса следующей записи в цепи применения (SD#ADNGS для MC или AD#ADNMS для TM)
RAUBASAI	a	Параметр указывает адрес поля в записи связи для адреса подчиненной базовой записи (SD#ADUBS для MC или AD#ADUBS для TM)
RAPRFUVI	a	Параметр указывает адрес контрольного поля в записи связи для подчиненной базовой записи (SD#PRUBS для MC или AD#PRUBS для TM)
LGUPRUFi	d	Параметром задается длина контрольного поля в записи связи для подчиненной базовой записи
RAPRFUBI	a	Параметр указывает адрес части поля GD#GIDNR или MD#MIDNR подчиненной базовой записи, которая должна выбираться из кода предмета или рабочего места для сравнения с контрольным полем в записях массива связей
RAESKETI	a	Параметр указывает адрес поля в подчиненной базовой записи для адреса первой записи из массива связей в цепи применения предмета или рабочего места (GD#ADEGS для МП или MD#ADEMS для PM)
LNGIDNRI	d	Параметром задается длина кода предмета или рабочего места в подчиненной базовой записи

Параметр	Операнд	Пояснения
RAVSKETI	d	Параметром указывается адрес поля в записи связи для адреса предыдущей записи в обратной цепи применения предметов или рабочих мест (SD#ADVGS для МС или AD#ADVMS для ТМ)
RASATZAI	a	Параметром указывается адрес поля в подчиненной базовой записи для счетчика записей в цепи применения предметов или рабочих мест (GD#GVZAE для МП или MD#MVZAE для РМ)
RAULAUF1	a	Параметром указывается адрес поля для текущего номера в записи МП
DADEFEND		Параметр указывает на конец записи параметров, относящихся к массиву связей
MAKROAUF	—	Параметр вызывает печать макрорасширений при трансляции модуля. Если параметр опущен, макрорасширения не печатаются

Таблица 4

Параметры генерации модуля DATEIORG для изменения массивов

Параметр	Операнд	Пояснения
PROGRART	DATEIORG	Параметром указывается тип модуля
PROGNAME	X	Параметром задается имя сгенерированного модуля в библиотеке объектных модулей (генерируется карта CATALR X)
BDHINZUF	—	Параметром предусматривается добавление записей в базовые массивы
BDLOESCH	—	Параметром предусматривается стирание записей в базовых массивах
CDDATNAM	X	Параметром задается имя МП, состоящее из семи знаков
BDAENDRN	—	Параметр задается, если предусматривается изменение МП
RALAUFNR	—	Параметр указывает адрес поля для текущего номера в записи МП (если задан LAUFNUMR)
DEFBADAT	—	Вслед за этим параметром необходимо записать параметры для изменяемого базового массива
DATEINAM	X	Параметром задается имя базового массива, состоящее из семи знаков
KETANKER	n	Параметром задается число цепей, начинающихся в базовом массиве. Рабочая цепь в МП в это количество не включается. В МП могут начинаться: цепь спецификаций; прямая цепь применения предметов; прямая цепь рабочих операций. В РМ может начинаться только прямая цепь использования рабочего места

Параметр	Операнд	Пояснения
RAANKER1	a	Параметром задается поле в записи для начального адреса первой цепи (GD#ADEPS для МП или MD#ADEMS для РМ)
RAANKER2	a	Параметром задается поле в записи для начального адреса второй цепи (GD#ADEGS)
RAANKER3	a	Параметром задается поле в записи для начального адреса третьей цепи (GD#ADEAS)
RAANKER4	a	Параметром задается поле в записи для начального адреса четвертой цепи (GD#ADLAS)
DADEFEND	—	Параметр указывает на окончание записи параметров, описывающих базовый массив. Аналогичным образом (от DEFBADAT до DADEFEND) описываются параметры для второго базового массива
VDLOESHE	—	Параметром предусматривается выполнение функции «стирание простое» для массивов связей
VDLOESHM	—	Параметром предусматривается выполнение функции «стирание групповое» для массивов связей
VDHINZUF	—	Параметром предусматривается выполнение функции «добавление» для массивов связей
VDBERICH	—	Параметром предусматривается выполнение функции «корректировка» для массивов связей
VERVDUBD	—	Параметром предусматривается выполнение функции «соединение» для ТМ и РМ
TREVDUBD	—	Параметром предусматривается выполнение функции «отделение» для ТМ и РМ
EINVEDAT	—	Параметр указывает на наличие в банке только одного массива связей. Если используются оба массива, параметр нужно опустить
SORTIER2	—	Параметр задается в случае, если в массивах связей используется второе поле сортировки (хотя бы в одном массиве)
SORTIER3	—	Параметр задается в случае, если в массивах связей используется третье поле сортировки (хотя бы в одном массиве)
DISPSTUF	—	Параметром предусматривается включение в систему программы вычисления ступеней диспозиции
RCKPRKET	—	Параметром предусматривается включение в систему программы создания обратной цепи рабочих операций
SATZZAEN	—	Параметром предусматривается резервирование поля счетчика записей в цепи. Поля счетчиков находятся в записях базовых массивов
EINUNTB	—	Параметр задается в случае, когда для массива связей имеется подчиненный базовый массив
KEINEUBD	—	Параметр задается в случае, когда ни один из массивов связей не имеет подчиненного базового массива

Параметр	Операнд	Пояснения
VORSEKET	—	Параметром предусматривается включение в систему программы создания прямой цепи применения предметов или рабочих мест
RCKSEKET	—	Параметром предусматривается включение в систему программы создания обратной цепи применения предметов или рабочих мест
DEFVEDAT	—	Вслед за этим параметром необходимо записать параметры для изменяемого массива связей
DATEINAM	X	Параметром задается имя массива связей, состоящее из семи знаков
WOSORTF1	V или B	Параметр указывает положение первого поля сортировки (V—в массиве связей, B—в базовом массиве)
RASORTF1	a	Параметром указывается относительный адрес первого поля сортировки в записи связей или в базовой записи
LGSORTF1	d	Параметром задается длина первого поля сортировки
WOSORTF2	V или B	То же для второго поля сортировки
RASORTF2	a	
LGSORTF2	d	
WOSORTF3	V или B	То же для третьего поля сортировки
RASORTF3	a	
LGSORTF3	d	
LGIDNRGD	d	Параметром задается длина кода предмета в записи МП
RAEPRKET	a	Параметром резервируется поле GD#ADEFS для MC или GD#ADEAS для TM в записи МП
RALPRKET	a	Параметром резервируется поле GD#ADLAS в записи МП
RAVPRKET	a	Параметром резервируется поле AD#ADVAS в записи TM. Параметр задается или вместе с предыдущим параметром, или оба опускаются
RASATZAE	—	Параметром резервируется поле в записи МП для счетчика записей в цепи (GD#POZAE для MC или GD#AGZAE для TM)
RAOLAUFN	a	Параметром резервируется поле для текущего номера в записи МП (GD#LAUFN)
RAPRUF0B	a	Параметром указывается адрес части кода предмета, используемой для сравнения с контрольным полем старшей базовой записи в записи связи. Эта часть перекрывает поле GD#GIDNR в записи МП
LGOPRUEF	d	Параметром задается длина поля, указанного предыдущим параметром
RAPRUF0V	a	Параметром задается контрольное поле для старшей базовой записи, расположенное в записи связи (SD#PROBS или AD#PROBS)
RAOBASAT	a	Параметром указывается адрес поля в записи связи для адреса старшей базовой записи (SD#ADOBS или AD#ADOBS)

Параметр	Операнд	Пояснения
RANPRKET	a	Параметром указывается адрес поля в записи связи для адреса следующей записи связи, относящейся к этой же старшей базовой записи (SD#ADNPS для MC или AP#ADNAS для TM)
RADSTUFE	a	Параметром указывается адрес поля в записи МП для ступеней диспозиции (GD#DISPO). Если ступени диспозиции не используются, этот и два следующих параметра нужно опустить
RAARBKET	a	Параметром указывается адрес поля в записи МП для адреса следующей записи в рабочей цепи (GD#ADNAK)
RAARKEPR	a	Параметром указывается адрес контрольного поля в записи МП для следующей записи в рабочей цепи (GD#AKPRU). Длина поля уже задана параметром LGOPRUEF (см. выше)
ANZUNTBD	1 или 0	Параметром указывается число подчиненных базовых массивов. Для TM задается значение 0, если PM отсутствует; в этом случае все следующие параметры до DADEFEND нужно опустить
RANSKETI	a	Параметром указывается адрес поля в записи связи для адреса следующей записи в прямой цепи применения предмета (SD#ADNGS) или рабочего места (AD#ADNMS)
RAUBASAI	a	Параметром указывается адрес поля в записи связи для адреса подчиненной базовой записи (SD#ADUBS или AD#ADUBS)
RAPRFUVI	a	Параметром указывается адрес контрольного поля в записи связи для подчиненной базовой записи (SD#PRUBS или AD#PRUBS)
LGUPRUF1	d	Параметром задается длина поля, указанного предыдущим параметром
RAPRFUBI	a	Параметром указывается адрес части кода предмета или рабочего места в подчиненной базовой записи, выбираемой для сравнения с контрольным полем в записи связи. Эта часть должна перекрывать поле GD#GIDNR или MD#MIDNR
RAESKETI	a	Параметром указывается адрес поля в записи базового массива для адреса первой записи в цепи применения предмета (GD#ADEGS) или рабочего места (MD#ADEMS)
LNGIDNRI	d	Параметром задается длина кода предмета или рабочего места в подчиненной базовой записи
RAVSKETI	a	Параметром указывается адрес поля в записи связи для адреса предыдущей записи в обратной цепи применения предмета (SD#ADVGS) или рабочего места (AD#ADVMS)
RASATZAI	a	Параметром указывается адрес поля в записи базового массива для счетчика записей в цепи применения предмета (GD#GVZAE) или рабочего места (MD#MVZAE)

Параметр	Операнд	Пояснения
RAULAUF1	a	Параметром указывается адрес поля для текущего номера в записи МП. Параметр задается только для МС
DADEFEND	—	Параметр указывает на конец записи параметров, описывающих массив связей. Аналогичным образом нужно записать параметры для следующего массива связей: от DEFVEDAT до DADEFEND
MAKROAUF	—	Параметр вызывает печать макрорасширений при трансляции модуля. Если параметр не задается, макрорасширения печататься не будут

Таблица 5

Параметры для генерации модулей ввода-вывода

Параметр	Операнд	Пояснения
PROGRART	EAWURZEL	Параметром задается тип модуля
PROGNAME	X	Параметром задается имя сгенерированного модуля в библиотеке объектных модулей (генерируется карта CATALR X)
MAKROAUF	—	При задании параметра распечатка Ассемблера будет выдаваться с макрорасширениями
ARPHANAM	X	Параметром указывается имя модуля EAARBEIT
UEBERLAG	d	Параметром задается размер в байтах области для вызова фаз EAARBEIT и EAEROABS. Сначала генерируются эти модули, и по распечатке Ассемблера определяется их максимальная длина в байтах
DEFBADAT	—	Вслед за этим параметром нужно записать параметры для базового массива, кончая параметром DADEFEND
DATEINAM	X	Параметром задается имя базового массива, состоящее из семи знаков
LNGEIDNR	d	Параметром указывается длина кода предмета или рабочего места
LNGESATZ	d	Параметром указывается длина записи базового массива, включая системную часть и часть пользователя
BLOCKFAK	n	Параметром задается число записей в блоке для базового массива
RAKENNZF	a	Параметром указывается адрес поля отметок в базовой записи (см. параметр RAKENNZF в табл. 2 настоящего приложения)

Параметр	Операнд	Пояснения
INDEXBER	d	Параметром задается длина области а памяти для считывания главного индекса или индекса цилиндров: $d_{min} = (z+1)(l+4)$, где z —число дорожек для индекса цилиндров, если параметр ZYLINDEX не задан, или число цилиндра для области данных, если параметр ZYLINDEX задан; l —длина кода предмета или рабочего места
DADEFEND	—	Параметром указывается конец записи параметра для одного базового массива. Аналогичным образом необходимо записывать параметры для следующего базового массива: от DEFBADAT до DADEFEND
DEFVEDAT	—	Вслед за этим параметром нужно записать параметры для массива связей, кончая параметром DADEFEND
DATEINAM	X	Параметром задается имя массива связей, состоящее из семи знаков
LNGEIDNR	d	Параметром задается длина кода предмета для МС. Для ТМ длина операнда должна быть равна нулю ($d=0$)
LNGESATZ	d	Параметром указывается длина записи массива связей, включая системную часть и часть пользователя
BLOCKFAK	n	Параметром указывается число записей в блоке для массива связей
DADEFEND	—	Параметр указывает на конец записи параметров для массива связей. Аналогичным образом нужно задать параметры для следующего массива связей: от DEFVEDAT до DADEFEND
PROGRART	EAARBEIT	Параметром задается тип модуля
PROGNAME	X	Параметром задается имя модуля. Оно должно совпадать с именем, заданным в параметре ARPHANAM (см. выше)
MAKROAUF	—	При задании параметра в распечатке Ассемблера будут печататься макрорасширения
EAERPHANAM	X	Параметром указывается имя модуля EAEROABS
BDAENDNR	—	Параметр указывается в случае, если этот модуль будет использоваться при изменении базовых массивов. Следующий параметр в этом случае использовать не разрешается
ZYLINDEX	—	Параметр задается в случае, если при поиске записей в базовом массиве для ускорения поиска индекс цилиндра должен загружаться в область памяти, зарезервированную параметром INDEXBER. Параметр нельзя использовать вместе с предыдущим. Рекомендуется употреблять этот параметр в модуле, сгенерированном для использования в программах обработки данных
PROGRART	EAEROABS	Параметром указывается тип модуля

Параметр	Операнд	Пояснения
PROGNAME	X	Параметром задается имя модуля. Оно должно совпадать с именем, заданным в параметре EAPHANAM (см. выше)
MAKROAUF	—	При задании параметра в распечатке Ассемблера будут печататься макрорасширения
BDAENDRN	—	Параметр указывает, что модуль будет использоваться при изменениях базовых массивов. Следующий параметр в этом случае задавать нельзя
ZYLINDEX	—	Параметр задается в случае, если индекс цилиндров должен загружаться в главную память. Его нельзя задавать вместе с предыдущим параметром
DEFBADAT	—	Вслед за этим параметром нужно записать параметры для базового массива
DATEINAM	X	Параметром указывается имя базового массива, состоящее из семи знаков
DADEFEND	—	Параметр указывает конец описания базового массива. Три последних параметра нужно повторить для следующего базового массива
DEFVEDAT	—	Вслед за этим параметром нужно записать параметры для массива связей
DATEINAM	X	Параметром указывается имя массива связей, состоящее из семи знаков
DADEFEND	—	Параметр указывает на конец описания массива связей. Последних три параметра нужно повторить для следующего массива связей

Таблица 6

Параметры для генерации модуля BDAREORG

Параметр	Операнд	Пояснения
PROGRART	BDAREORG	Параметром указывается тип модуля
PROGNAME	X	Параметром задается имя модуля в библиотеке объектных модулей
EBDATNAM	X	Параметром указывается имя исходного (реорганизуемого) базового массива
BDAENDER	—	Параметр задается в случае, если при реорганизации предусматриваются изменения в базовых массивах (добавление, корректировка, стирание, изменение структуры записей). Изменения выполняются с помощью СПП32
EBLGSATZ	d	Параметром указывается длина записей исходного массива, если предусмотрены изменения

Параметр	Операнд	Пояснения
BDALADEN	—	Параметр задается в случае, если результирующий (реорганизованный) массив должен загружаться на диск. Если загрузка не требуется, этот и четыре следующих параметра опускаются. Для вывода реорганизованного массива в этом случае нужно использовать СППЗЗ
ABDATNAM	X	Параметром задается имя результирующего массива, состоящее из семи знаков. Если это имя совпадает с именем исходного массива, массив будет загружен на прежнее место на диске. Для загрузки на новое место нужно задать другое имя. Это имя используется только данным модулем и не влияет на другие модули или программы. Однако, если параметры массива изменились, зависящие от этих параметров модули и программы необходимо генерировать заново
LNGESATZ	d	Параметром задается длина записи результирующего массива
BLOCKFAK	n	Параметром задается количество записей в блоке для результирующего массива
RAKENNZF	a	Параметром указывается адрес поля для отметок в записях результирующего массива
LNGEIDNR	d	Параметром указывается длина кода предмета или рабочего места в результирующем массиве. Она должна совпадать с длиной кода в исходном массиве
ARBDATFI	X	Параметром задается имя рабочего массива для промежуточного накопления записей при загрузке массива на прежнее место. Имя должно состоять из семи знаков
ARBDABER	n или 5	Параметром указывается число несвязанных областей на диске для рабочего массива. Если число не превышает пять, параметр можно опустить. Размеры массива должны быть такими, чтобы он мог вместить все записи, добавленные с момента загрузки или последней реорганизации, в противном случае реорганизация прекращается, и массив оказывается испорченным
ARBDLKF	n	Параметром задается число записей в блоке для рабочего массива. Если параметр опущен, принимается число, заданное для результирующего массива
VOLLSPUR	—	Если задан этот параметр, блок записей в рабочем массиве будет занимать целую дорожку. Параметр можно задавать вместо предыдущего для ускорения обработки, однако в этом случае увеличится потребность в памяти для области ввода-вывода

Параметр	Операнд	Пояснения
VDAENDER	—	Параметром указывается, что для базового массива имеется хотя бы один массив связей. Если массивов связей нет, этот параметр и последующие, включая DADEFEND, опускается
DEFVEDAT	—	Вслед за этим параметром нужно задать параметры для массивов связей, описывающие одну цепь, начинающуюся в базовом массиве
VDDATNAM	X	Параметром указывается имя массива связей, через который проходит цепь
RAPRUEFB	a	Параметром указывается адрес части кода предмета или рабочего места, выбираемой для сравнения с контрольным полем в записи связи для этой цепи
LGPRUEFB	d	Параметром указывается длина контрольного поля
RAANKERB	a	Параметром указывается адрес поля в базовой записи для адреса первой записи в цепи
RALVADRB	a	Параметром указывается адрес поля в записи МП (GD#ADLAS) для адреса обратной цепи рабочих операций. Этот параметр записывается в последовательности параметров DEFVEDAT — DADEFEND только в том случае, если параметром RAANKERB указан адрес поля GD#ADEAS и имеется обратная цепь рабочих операций
RAZAEHLB	a	Параметром указывается адрес поля счетчика записей в цепи
RABDADRV	a	Параметром указывается в записи связи адрес поля, содержащего адрес базовой записи: SD#ADOBS — для цепи спецификаций; SD#ADUBS — для цепи применения предметов; AD#ADOBS — для цепи рабочих операций; AD#ADUBS — для цепи использования рабочих мест
RAPRUEFV	a	Параметром указывается адрес контрольного поля в записи связи: SD#PROBS — для цепи спецификаций; SD#PRUBS — для цепи применения предметов; AD#PROBS — для цепи рабочих операций; AD#PRUBS — для цепи использования рабочих мест
RANVADRV	a	Параметром указывается адрес поля в записи связи для адреса следующей записи в цепи
RAVVADRV	a	Параметром указывается адрес поля в записи связи для адреса предыдущей записи в цепи (если имеется обратная цепь)
DADEFEND	—	Параметр указывает на конец параметров описания одной цепи. Аналогичные последовательности параметров от DEFVEDAT до DADEFEND нужно задать для остальных цепей, начинающихся в этом базовом массиве. Для обратной цепи рабочих операций отдельная последовательность не требуется (см. параметр RALVADRB)

Параметр	Операнд	Пояснения
LNGEVMAX	d	Параметром указывается максимальная длина записей связей
DFKONSTF	—	Вслед за этим параметром нужно записать на языке Ассемблера константы и поля, используемые в СППЗ1—СППЗ4 (см. гл. 1, §7)
DEFINEND	—	Параметр указывает на конец записи констант и полей
ANSAROnn	—	Вслед за этим параметром на языке Ассемблера нужно записать команды СПП (см. описание СППЗ1—СППЗ4, гл. 1, §7)
ENSAROnn	—	Параметр указывает на конец записи СПП. Если требуются другие СПП, их нужно оформить аналогичным образом
DFANWDAT	—	Вслед за этим параметром нужно запрограммировать описание дополнительных массивов, используемых при реорганизации, если таковые имеются (с помощью макрокоманд DTF)
DEFINEND	—	Параметр указывает на конец описания дополнительных массивов
DFEINBAD	—	Вслед за этим параметром можно задать структуру системной части и части пользователя в записи исходного базового массива, если это требуется для СПП
DEFINEND	—	Параметр указывает на конец описания структуры записи. Если структуру записи описывать не нужно, оба параметра опускаются. Исходная запись при реорганизации помещается в поле с именем BR#ESABR, префикс записи — в поле BR#EARBR и код предмета или рабочего места — в поле BR#ESCHL
DFAUSBAD	—	То же для результирующей записи. Запись находится в поле BR#ASABR, префикс — в поле BR#AARBR и код — в поле BR#ASCHL
DEFINEND	—	
MAKROAUF	—	При задании параметра в распечатке Ассемблера будут печататься макрорасширения

Таблица 7

Параметры для генерации модуля VDAREORG

Параметр	Операнд	Пояснения
PROGRART	VDAREORG	Параметром задается тип модуля
PROGNAME	X	Параметром задается имя сгенерированного модуля в библиотеке объектных модулей (генерируется карта CATALR X)
PHASE002	—	Параметр задается в случае, если в массиве связей имеются вторичные цепи (цепи применения) или же их необходимо создать в процессе реорганизации

Параметр	Операнд	Пояснения
PHASE003	—	Параметром указывается, что в подчиненном базовом массиве необходимо корректировать начальные адреса вторичных цепей. Для ТМ параметр можно опустить, если он не связан с РМ
NURPHAS3	—	При задании параметра модуль будет содержать только фазу 3 и выполнять только корректировку начальных адресов в подчиненном базовом массиве
DISPSTUF	n	Если задается этот параметр, записи в МС после реорганизации будут расположены по возрастанию ступеней диспозиции, а для одинаковых ступеней диспозиции по возрастанию кодов сборочных единиц. Максимальное число ступеней задается операндом. Если число ступеней диспозиции в массиве превысит заданное, произойдет программный сбой
OBDATNAM	X	Параметром задается имя старшего базового массива, состоящее из семи знаков
LGIDNROB	d	Параметром указывается длина кода предмета в записи старшего базового массива
LGSATZOB	d	Параметром указывается длина записи старшего базового массива
RAEPRKET	a	Параметром задается в МП адрес поля, содержащего начальный адрес первичной цепи: GD#ADEPS — для МС или GD#ADEAS — для ТМ
RALPRKET	a	При наличии обратной цепи рабочих операций (GD#ADLAS) с помощью данного параметра указывается адрес поля в записи МП
RADSTUFE	a	Параметром задается адрес поля в записи МП для ступени диспозиции (GD#DISPO). Этот параметр и два следующих опускаются, если не был задан параметр DISPSTUF
RAARBKET	a	Параметром задается адрес поля в записи МП для адреса записи в рабочей цепи (GD#ADNAK)
RAARKEPR	a	Параметром задается адрес контрольного поля в записи МП для рабочей цепи (GD#AKPRU)
RALAUFNR	a	Параметром задается адрес поля в записи МП для текущего номера
RAPRUFOB	a	Параметром задается адрес поля в записи старшего базового массива, выбираемого для сравнения с контрольным полем старшей базовой записи в записи связи
LGPRUFOB	d	Параметром задается длина поля, указанного предыдущим параметром
EVDATNAM	X	Параметром задается имя исходного (реорганизуемого) массива связей, состоящее из семи знаков
AVDATNAM	X	То же для результирующего (реорганизованного) массива связей
LGSATZEV	d	Параметром задается длина записи исходного массива связей

Параметр	Операнд	Пояснения
BLOCKFAK	n	Параметром задается число записей в блоке для реорганизованного массива связей
LGSATZAV	d	Параметром задается длина записи реорганизованного массива связей
KEINEUBD	—	Параметр указывает, что подчиненный базовый массив отсутствует. Параметр относится только для ТМ, если он не связан с РМ. Последующие параметры до RAPRFOV1 нужно в этом случае опустить
UEBUNTGL	—	Параметр указывает, что старший и подчиненный базовые массивы совпадают. Параметр задается для МС, для ТМ опускается
UBDATNAM	x	Параметром задается имя подчиненного базового массива, состоящее из семи знаков
LGIDNRUB	d	Параметром задается длина кода предмета или рабочего места в подчиненном базовом массиве
LGSATZUB	d	Параметром задается длина записи подчиненного базового массива
RAPRUFUB	a	В записи подчиненного базового массива параметром задается адрес поля, предназначенного для сравнения с контрольным полем подчиненной базовой записи в записи связи
LGPRUFUB	d	Параметром задается длина поля, указанного предыдущим параметром
RAESKETT	a	Параметр указывает адрес поля GD# #ADEGS в записи МП для МС или MD# #ADEMS в записи РМ для ТМ. Параметр задается в случае, если был задан параметр PHASE002 или NURPHAS3 и если имеется цепь применимости предметов или рабочих мест
RAUBASA1	a	Параметром задается адрес поля SD# #ADUBS для МС или AD#ADUBS для ТМ. Этот и следующий параметры задаются, если был задан параметр PHASE003
RAPRFUV1	a	Параметром задается адрес поля SD# #PRUBS для МС или AD#PRUBS для ТМ
RAPRFOV1	a	Параметром задается адрес поля SD# #PROBS для МС или AD#PROBS для ТМ
RAOBASA2	a	Параметром задается адрес поля SD#ADOBS для МС или AD#ADOBS для ТМ
RAPRFOV2	a	Параметром задается адрес поля SD# #PROBS для МС или AD#PROBS для ТМ
RANPRKT2	a	Параметром задается адрес поля SD#ADVAS для МС или AD#ADVAS для ТМ
RAVPRKT2	a	Параметром задается адрес поля AD#ADVAS для ТМ, если имеется обратная цепь рабочих операций
RAUBASA2	a	Параметром задается адрес поля в записи связи для адреса подчиненной базовой записи. Этот и следующий параметр опускается, если не были заданы параметры PHASE002 или PHASE003

Параметр	Операнд	Пояснения
RAPRFUV2	a	Параметром задается адрес поля SD# #PRUBS для MC или AD#PRUBS для TM
DEFVEDAT	—	Вслед за этим параметром необходимо записать параметры, описывающие вторичную цепь — цепь применения предметов или рабочих мест. Если эта цепь отсутствует, необходимо опустить этот и последующие параметры до DFKONSTF. Опускаются параметры также в случае, если не был задан PHASE002
RANSKET2	a	Параметром задается адрес поля SD#ADNGS для MC или AD#ADNMS для TM, если используется цепь применяемости предметов или рабочих мест
RAVSKET2	a	Параметром задается адрес поля SD# #ADVGS для MC или AD#ADVMS для TM, если имеется обратная цепь применяемости предметов или рабочих мест
DADEFEND	—	Данный параметр является признаком конца описания параметров для вторичной цепи
DFKONSTF	—	Вслед за этим параметром необходимо записать константы и поля для СПП на языке Ассемблера. Описание полей и констант необходимо закончить параметром DEFINEND без операнда
DFUEBBAD	—	Вслед за этим параметром необходимо описать структуру записей старшего базового массива, если в СПП43 предусмотрена печать табуляграммы с использованием данных из старшей базовой записи. Если параметр опустить, старшую базовую запись можно брать из области с именем VR#BSABR. Описание структуры записи нужно закончить параметром DEFINEND без операнда
DFEINVED	—	Вслед за этим параметром нужно описать структуру записи исходного массива связей, если это требуется для СПП43. Если параметр опустить, запись связи можно брать из области с именем VR#ESABR. Закачивается описание параметром DEFINEND
DFAUSVED	—	Вслед за этим параметром нужно описать структуру записи реорганизованного массива связей, если это требуется для СПП43. Если параметр опустить, запись можно брать из области с именем VR#ASABR. Закачивается описание параметром DEFINEND
VDAENDER	—	Параметр указывает на то, что при реорганизации массива связей предусмотрена дополнительная обработка записей и обращение к СПП43

Параметр	Операнд	Пояснения
DFANWDAT	—	Вслед за этим параметром следует описать дополнительные массивы, используемые при реорганизации (если они имеются) с помощью макрокоманд DTF и закончить описания параметром DEFINEND без операнда
ANSAROnn	—	Вслед за этим параметром нужно написать СИПnn на языке Ассемблера (nn принимает значения 41—43)
ENSAROnn	—	Параметр является признаком конца записи СИПnn. Каждая СИП должна начинаться параметром ANSAROnn и заканчиваться ENSAROnn
MAKROAUF	—	При задании параметра в распечатке Ассемблера печатаются макрорасширения

3. УПРАВЛЯЮЩИЕ КАРТЫ ДЛЯ ГЕНЕРАЦИИ МОДУЛЕЙ

```
// JOB ANPASSEN
// OPTION LINK
// ASSIGN SYSRLB,X'191'
// DLBL IJSYRL,'SOPS'/BA/OBJ'
// INCLUDE HIP#AN
// INCLUDE HIP#WU
// INCLUDE HIP#AR
// INCLUDE HIP#OS
// LBLTYP NSD(03)
// EXEC LNKEDT
// ASSIGN SYSPCH,X'280'
// MTC REW,SYSPCH
// ASSIGN SYS006,X'191'
// DLBL DB#PROG,'ALLGEMEINES PROBLEMPROGRAMM',99/365,DA
// EXTENT SYS006,BASTE1,1,0,10,1
// EXTENT SYS006,BASTE1,1,1,11,2
// EXTENT SYS006,BASTE1,1,2,20,470
// DLBL DB#ANBR,'ANPASSUNGSBEREICH',01/001,DA
// EXTENT SYS006,BASTE1,1,0,490,400
// EXEC
карты параметров
/*
/&
```

Полученные в процессе генерации модули выводятся на магнитную ленту X'280'. Эту ленту необходимо назначить в качестве устройства SYSIPT для Ассемблера.

4. УПРАВЛЯЮЩИЕ КАРТЫ ДЛЯ ИЗМЕНЕНИЯ БАНКА ДАННЫХ

```
// JOB AENDERN
// OPTION LINK
// ASSIGN SYSRLB,X'191'
```

```
// DLBL IJSYSRL,'SOPS/BA/OBJ'
// INCLUDE MOD1
// INCLUDE MOD2
// INCLUDE MOD3
// INCLUDE MOD4
// INCLUDE MOD5
// LBLTYP NSD(03)
// EXEC LNKEDT
// ASSGN SYS008,X'192'
// ASSGN SYS010,X'280'
// MTC REW,SYSIPT
// DLBL MASSIV1,'МП',99/365,DA
// EXTENT SYS005,1,0,10,1
// EXTENT SYS005,1,1,11,3
// EXTENT SYS005,1,2,20,100
// DLBL MASSIV2,'МС',99/365,DA
// EXTENT SYS005,1,0,120,200
// DLBL MASSIV3,'ПМ',99/365,DA
// EXTENT SYS005,1,0,14,2
// EXTENT SYS005,1,1,16,4
// EXTENT SYS005,1,2,220,100
// DLBL MASSIV4,'ТМ',99/365,DA
// EXTENT SYS005,1,0,320,500
// EXEC
/&
```

главный индекс
индекс цилиндров
область данных
область данных

В примере сделаны следующие предположения. Библиотека объектных модулей расположена на устройстве X'191', массивы банка данных — на устройстве X'192', входной массив для изменений — на МЛ X'280'.

Модули в библиотеке имеют имена:

```
LEITPROG — MOD1;
DATEIORG — (для изменений) — MOD2;
EAWURZEL — MOD3;
EAARBEIT — MOD4;
EAEROABS — MOD5.
```

Как видно из примера, для базовых массивов нужно задавать по три участка памяти на дисках, для массивов связей — по одному.

5. ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ ЯЗЫКА ПРОЦЕДУР СИСТЕМЫ САВИ

А. Инструкции

DEFINIEREN	— определение
LADEN	— загрузка
DATEI	— массив
BEREICH	— область
IDENT	— идентификатор записи
SEGMENT	— сегмент
FELD	— поле
SYNONYME	— синонимы
TEXT	— текстовая часть
TESTEN	— проверить
KODIEREN	— кодировать
DEKODIEREN	— декодировать
AUFBEREITEN	— подготовить

SPEICHERN
 ERWEITERN
 ERSETZEN
 BERICHTIGEN
 ZUFUEGEN
 LOESCHEN
 STILLEGEN
 AKTIVIEREN
 AENDERN
 ZUORDNEN
 EINFUEGEN
 SUCHEN
 WENN
 AUSGEBEN
 DRUCKEN
 STANZEN
 ZAEHLEN
 ADDIEREN
 SORTIEREN
 TITEL
 KOPF
 FUSS
 VORSCHUB
 ZEILE

— накопить
 — расширить
 — заменить
 — исправить
 — добавить
 — стереть
 — консервировать
 — активизировать
 — изменить
 — подчинить
 — включить
 — поиск
 — условие
 — вывод
 — печать
 — перфорация
 — счет
 — сложение
 — сортировать
 — заголовок
 — шапка
 — последняя строка
 — сдвиг
 — строка

Б. Ключевые слова

TYP
 SATZART
 SATZFORMAT
 SATZLAENGE
 UEBERLAUFSP
 BLOCKLAENGE
 BLOCKFAKTOR
 ZUCODE
 ANF
 LAENGE
 SATZ
 ZUSTAND
 FOLGE
 FUNKTION
 SEGMENT
 SATZENDE
 FORMAT
 FELDENDE
 SEGMENTENDE
 NAME
 DARST
 MASKE
 BEREICH
 WERTE
 PHASE
 DATEN
 TAFEL

— тип массива
 — вид записей
 — формат физических записей
 — длина записи внутри массива
 — число дорожек переполнения в цилиндре
 — длина блока
 — число записей в блоке
 — код, защищающий массив
 — начало области
 — длина области
 — запись
 — состояние
 — последовательность
 — поле функции
 — поле сегмента
 — знак конца логической записи
 — формат сегмента
 — конец поля
 — конец сегмента
 — имя (присваивается полю)
 — форма представления
 — контрольная маска
 — область
 — значение
 — фаза
 — данные
 — таблица

В. Зарезервированные слова

ES
 MS
 FEST
 TRZN
 IMPL

— однократная запись
 — многократная запись
 — фиксированный формат
 — формат разделителей
 — подчинение неявное

NUR	— только
AUSSER	— кроме
EXPL	— подчиненное явное
ALLE	— все
AUS	— из
BEI	— при
MIT	— с
VOREIN	} точки выхода при накоплении и поиске
MODEIN	
VORAUSS	—
NICHT	— нет
UND	— и
ODER	— или
CLEICH	— равно
GROESSER	— больше
KLEINER	— меньше
DURCHSUCHEN	— просмотреть
ERFUELLT	— выполнено
AUF	— на
PLATTE	— диск
BAND	— лента
KBAND	— магнитная лента с метками
PARALLELDRECKER	— устройство параллельной печати
KARTE	— перфокарта
LOCHBAND	— перфолента
IN	— в
EXTERN	— внешняя
INTERN	— внутренняя
HEXADEZIMAL	— 16-ричная
DEZIMAL	— десятичная

6. ЯЗЫК ОПИСАНИЯ ДАННЫХ ДЛЯ СОЗДАНИЯ ТАБЛИЦ ОПИСАНИЯ ДАННЫХ (ТОД)

При создании ТОД определяется структура массивов и записей массивов (входных, выходных), управляемых и обрабатываемых с помощью САВИ. При выполнении программы анализируются инструкции языка описания данных и составляются соответствующие ТОД. Таблица описания данных, содержащая всю информацию о структуре записи и массива, является необходимым условием для выполнения функций накопления и поиска системы программирования САВИ.

Для создания ТОД используются следующие инструкции:

DEFINIEREN — задать имя ТОД. Формат инструкции:

DEFINIEREN имя ТОД _;

где имя ТОД — 1—6 алфавитно-цифровых знаков (первый знак всегда алфавитный);

LADEN — загрузить. Эта инструкция используется для загрузки ТОД. Формат инструкции:

LADEN _;

ERSETZEN — заменить. Инструкция используется при создании ТОД в том случае, если необходимо исправить содержание уже имеющейся ТОД. Формат инструкции:

ERSETZEN _;

DATEI — массив. Формат инструкции:

```
[имя] DATEI [TYP]
               [SATZART=]
               [SATZFORMAT=]
               [SATZLAENGE=]
               [UEBERLAUFSP=]
               [ [ BLOCKLAENGE= ] ]
               [ BLOCKFAKTOR= ] ]
               [ZUCODE=]
               [SPCODE=];
```

где операнд TYP описывает тип массива; если массив на перфоносителях или на магнитных носителях и подготовлен средствами, не принадлежащими системе CABI, то необходимо указать параметр PRIMAER, если массив подготовлен на магнитных носителях с помощью CABI, то — SEKUNDAER. Если операнд не указан, принимается SEKUNDAER;

операнд SATZART описывает вид записей массива и используется только для описания входного массива на перфокартах. Если логическая запись состоит из одной физической записи (ПК), то указать ES, если логическая запись состоит из нескольких физических записей (не более пяти), то указать MS,n; (n — число физических записей (ПК));

операнд SATZFORMAT описывает формат физической записи массива. Для указания формата используются параметры FEST — фиксированный, VARIABLE — переменный и UNDEFINIERT — неопределенный. Если операнд не указан, то принимается фиксированный формат;

операнд SATZLAENGE=nnnn описывает длину записи массива. Максимальное значение nnnn=3600 байт;

операнд UEBERLAUFSP=n указывает возможное число дождек переполнения для индексно-последовательных массивов; максимальное значение равно 8, если операнд опущен, то оно принимается равным 2;

операнд BLOCKLAENGE=nnnnn описывает длину блока массива в байтах. Максимальное значение длины блока массива на диске равно 3600 байт, на МЛ — 32 767 байт;

операнд BLOCKFAKTOR=nnn описывает длину блока массива в записях. Использование операндов BLOCKLAENGE и BLOCKFAKTOR для описания длины блока массива в одной ТОД запрещается;

операнд ZUCODE описывает код выборки, разрешающий доступ к массиву;

операнд SPCODE описывает код защиты, разрешающий вести накопление массива. В качестве кода выборки и кода защиты используется сочетание из десяти любых печатных знаков;

BEREICH — область. Эта инструкция используется только для описания областей, включающих ведущие реквизиты и информационные реквизиты на перфокартах. Ведущие реквизиты — рекви-

зтиты-признаки, однозначно определяющие документостроку (запись) в массиве. Область, описывающая ведущие реквизиты, определяется как идентификационная область. В нее могут входить, кроме ведущих реквизитов, любые служебные реквизиты. Идентификационная область может состоять из одной подобласти или из двух подобластей, но не более, и может располагаться только в начале записи или в конце записи, или в начале и в конце записи (случай с двумя подобластями). Формат инструкции:

[имя] BEREICH ANFANG=
LAENGE=

где операнд ANFANG=пп описывает начало области, номер позиции, с которой начинается область, пп=1—80;

операнд LAENGE=пп описывает длину области, число позиций, занимаемых областью.

Если идентификационная область состоит из двух подобластей, то операнды записываются следующим образом:

ANFANG=пп,пп
LAENGE=пп,пп

здесь первые пп указывают длину первой подобласти, вторые пп — длину второй подобласти;

IDENT — идентификатор записи (ведущие реквизиты). Формат инструкции:

[имя] IDENT SATZ=
[ZUSTAND=]
[FUNKTION=]
[FOLGE=]
[SEGMENT=]
[SATZENDE=];

где операнд SATZ описывает имена полей идентификационной области в порядке рассортированности массива. Можно указать до 15 имен полей. Имена полей отделяются друг от друга запятой.

Имя поля не должно превышать 25 алфавитно-цифровых знаков;

операнд ZUSTAND (только для основных массивов) описывает поле состояния, содержание которого характеризует, какую работу необходимо проделать с записью массива. Длина поля состояния не должна превышать один байт;

операнд FUNKTION описывает поле функции (только для входных массивов), в котором задается признак корректировки массива. Длина поля функции не должна превышать один байт;

операнд FOLGE описывает поле последовательности, содержание которого указывает на порядок следования физических записей (ПК) в логической записи. Длина поля последовательности не должна превышать один байт;

операнд SEGMENT описывает идентификационное поле сегмента;

операнд SATZENDE задает конечный признак логической записи. Признак может быть представлен в форме:

единичного знака — Z;
 знаковой константы — 'Z';
 шестнадцатеричной константы — X'w...w';
 двоичной константы — B'mmmmmmmmm';
 SEGMENT — сегмент. Формат инструкции:

```
[имя] SEGMENT [FORMAT=]
                  [FELDENDE=]
                  [SEGMENTENDE=]
                  [IDENT=]
```

где операнд FORMAT описывает формат сегмента. Для описания формата сегмента используются параметры:

FEST — фиксированный формат, TRZN — формат разделителей, TEXT — текст, IDENT, n, y — формат ключевого слова. (Здесь n — длина (1—4 байта) признака идентификационного поля, y — разделитель идентификационного поля.) Если операнд опущен, то принимается фиксированный формат;

операнд FELDENDE=Z задает признак конца поля (здесь Z — любой печатный знак);

операнд SEGMENTENDE=Z задает признак конца сегмента (здесь Z — любой печатный знак, не совпадающий с признаком конца поля);

операнд IDENT описывает признак идентификатора сегмента. Признак задается как последовательность от 1 до 5 любых печатных знаков в виде:

UUUUU — цепь печатных знаков;
 'UUUUU' — последовательность знаковой константы;
 X'W...W' — последовательность шестнадцатеричной константы;
 B'mmmmmmmmm' — последовательность двоичной константы;

FELD — поле. Формат инструкции:

```
[имя] FELD [NAME=]
              [LAENGE=]
              [DARST=]
              [IDENT=];
```

где операнд NAME задает имя поля. Имя поля задается последовательностью от 1 до 25 алфавитно-цифровых знаков; операнд LAENGE=nnn задает длину поля (здесь nnn — значение от 1 до 256 байт в зависимости от формы представления);

операнд DARST=W определяет форму представления содержимого поля (здесь W — признак формы представления: F — фиксированное число, P — десятичное упакованное число, T — алфавитно-числовой код, N — десятичное неупакованное число, V — цифровой со знаком числа, A — двоичный абсолютный код, X — шестнадцатеричный код, B — двоичный код, D — дата (упакованное число). Если операнд опущен, то принимается алфавитно-числовой код T;

операнд IDENT определяет идентификационный признак поля (укороченный ключ) для полей в сегментах формата IDENT.

Идентпризнак поля представляет собой последовательность от 1 до 4 печатных символов и может быть в виде:

UUUU — цепочки символов;
'UUUU' — знаковой константы;
X'W...W' — шестнадцатеричной константы;
B'm...m' — двоичной константы;

SYNONYME — синоним. Инструкция используется для описания синонимов имени поля. Формат инструкции:

[имя] SYNONYME NAME=;

TEXT — текст. Эта инструкция используется для описания чисто текстовых полей записи. Формат инструкции:

[имя] TEXT NAME=
[LAENGE=];

где операнд **NAME** описывает имя текстового поля;

операнд **LAENGE** описывает длину (от 1 до 3600 байт) поля. Если длина поля в инструкции не указывается, то она рассчитывается системой из длины записи;

TESTEN — проверить. Данная инструкция используется для проверки содержимого полей. Формат инструкции:

[имя] TESTEN { MASKE=
BEREICH=
WERTE=
PHASE=[EINGANG=][DATEN=]
DATEN= };

где операнд **MASKE** описывает контрольную маску согласно длине проверяемого поля для проверки полей на цифровое значение, на наличие конкретных знаков в определенных позициях поля;

операнд **BEREICH** описывает таблицу (область значений). Таблица может быть как внешней, так и внутренней. Во внешней таблице операнд описывает имя таблицы (1—8 алфавитно-цифровых знаков), которая хранится в виде фазы. Во внутренней таблице области значений задаются в операнде. Области значений записываются в таблице в виде: min1, max1, min2, max2,..., (здесь min — нижнее значение области, max — верхнее значение области). Можно указать не более пяти пар значений (областей значений). Значения могут быть представлены в виде цепи символов, знаковой константы, двоичной константы и цифровой величины;

операнд **WERTE** описывает таблицу значений. Таблица может быть как внешней, так и внутренней (см. **BEREICH**). Можно указать до 256 значений;

операнд **PHASE** описывает имя дополнительного модуля (модуля пользователя) для проверки входных полей при накоплении. Имя модуля может состоять из 1—8 алфавитно-цифровых знаков;

операнд **EINGANG** задает форму представления и длины входных полей для дополнительного модуля в виде;

EINGANG=W,nnn

здесь W — символ формы представления (см. операнд DARST);
 ppp — необходимая длина входного поля (десятичное число
 между 1 и 256, максимальное значение которого зави-
 сит от формы представления);

операнд DATEN задает данные пользователя для управления
 дополнительным модулем. Данные представляются в виде:

элемент 1,
 элемент 2,

Максимальное число элементов — 16 при общей длине в
 256 байт. Элементы могут задаваться в виде знаковой константы,
 цепочки символов, шестнадцатеричной константы и двоичной
 константы;

KODIEREN — кодировать. Инструкция используется для пере-
 кодирования содержимого полей в код, удобный для машинной
 обработки. Формат инструкции:

[имя] KODIEREN { PHASE=[EINGANG=][DATEN=]
 DATEN=
 TAFEL=[EXTERN=] };

где операнд PHASE описывает имя дополнительного модуля
 (см. описание инструкции TESTEN; там же — описание операнда
 DATEN);

операнд TAFEL задает таблицу для перекодирования. Таблица
 может быть как внешней, так и внутренней (см. описание опера-
 нда BEREICH). Значения таблицы задаются в виде:

аргумент 1, функция 1,
 аргумент 2, функция 2,

В таблице можно задавать до 128 пар значений;
 операнд EXTERN задает форму представления и длину зна-
 чений аргумента в таблице в виде w,ppp (см. операнд EINGANG);
 DEKODIEREN — декодировать. Формат инструкции:

[имя] DEKODIEREN { PHASE=[EINGANG=][DATEN=]
 DATEN=
 TAFEL=[EXTERN=] };

Для инструкции DEKODIEREN необходимо указывать те же
 самые операнды ключевых слов и параметры, что и при KODIE-
 REN, только операнд TAFEL может быть представлен в виде:

TAFEL = { имя внешней таблицы
 аргумент 1, функция 1 [аргумент 2, ...] };
 *.INVERS

здесь параметр *.INVERS употребляется в случае, если внутрен-
 нюю таблицу перекодирования в инструкции KODIEREN необходи-

мо будет использовать в обратном направлении при декодирования.

AUFBEREITEN — подготовить. Инструкция используется для подготовки содержимого полей перед выводом на печать. Формат инструкции:

$$[\text{нмя}] \text{ AUFBEREITEN } \left\{ \begin{array}{l} \text{PHASE} = [\text{EINGANG} =] [\text{DATEN} =] \\ \text{DATEN} = \\ \text{MASKE} = \end{array} \right\};$$

где операнд **MASKE** задает маску подготовки содержимого поля для вывода на печать в виде: маска, пп. Инструкция ограничивается 31 знаком и отличается от соответствующей инструкции Ассемблера только знаками, ее образующими. Для образования маски используются знаки: @ — символ указания начала значащих цифр (действительных); & — символ разделения поля; ? — символ указания цифр. Маска заключается в апострофы. Например, если шаблон реквизита 9(3), 9(2), то маска будет иметь следующий вид: '□? @?, ??'. Первый символ после апострофа описывает знак, заполняющий место подавленных нулей, пп — длина постоянной повторяющейся части шаблона, которая всегда указывается в выходном поле;

описание операндов **PHASE**, **EINGANG** и **DATEN** дано в инструкции **TESTEN**.

7. ЯЗЫК ОПЕРАЦИЙ ДЛЯ СОЗДАНИЯ И ВЕДЕНИЯ ОСНОВНЫХ МАССИВОВ

Создание и ведение основных массивов осуществляется под управлением инструкции **SPEICHERN** (накопить). Она вызывает секцию программ накопления и подготовки необходимых **ТОД** входного и основного массивов. Формат инструкции:

SPEICHERN нмя **ТОД** основной
 код накопления
AUS
 нмя **ТОД** входной
 код доступа к данным;

где код накопления — код пользователя, дающий право накапливать данные в соответствующем основном массиве. Путем сравнения заданного кода с кодом, занесенным в **ТОД**, выясняется, имеет ли право пользователь выполнять операции накопления. Если защита накопления не предусмотрена, вместо кода пользователь может задавать 4—20 любых знаков;

AUS — фиксированное слово, соединяющее информацию об основном массиве с информацией о входном массиве;

Код доступа к данным — код пользователя, дающий право обращаться к соответствующему входному массиву. Если защита доступа не предусмотрена, то можно задавать любые 4—20 знаков.

Пример инструкции:

SPEICHERN BAA011 AAAA AUS A01011 BBBB ;

Инструкция LADEN (загрузить) вызывает построение основного массива из входного массива. Инструкция состоит из одного фиксированного слова, имя инструкции не указывается. Формат инструкции:

LADEN__;

Инструкция ERWEITERN (расширить) вызывает дозапись новых записей в конец существующего массива. Формат инструкции:

ERWEITERN__;

Инструкция ERSETZEN (заменить) вызывает замену содержания некоторых полей или целых записей. Формат инструкции:

ERSETZEN__;

Данной инструкцией обрабатываются все подчиненные поля, поэтому она рассматривается в тесной связи с инструкцией ZUORDNEN (подчинить). Если необходимо заменить все основные записи, все поля должны быть подчиненными. При замене некоторых полей подчиненными должны быть лишь выбранные поля. Подчиненные поля, содержащие во входном массиве пробелы, получают в основном массиве нулевое значение, которое зависит от формы представления. Неподчиненные поля не изменяются.

Инструкция BERICHTIGEN (корректировать) вызывает изменение одного или нескольких реквизитов внутри одной основной записи. Формат инструкции:

BERICHTIGEN__;

При выполнении данной инструкции обрабатываются только те поля, для которых во входной записи задан реквизит. Если же при жестко форматированном вводе необходимо стереть поле с формой представления T, то в первом байте соответствующего поля входной записи задается знак %. При прочих формах представления заносится Ø. Если входные записи имеют формат разделителей или ключевого слова, знак Ø задавать не надо.

При корректировке может использоваться как входная ТОД, предназначенная для загрузки, так и ТОД, используемая для внесения особых изменений (например, может оказаться целесообразным переход от формата разделителей к формату ключевого слова).

Инструкция ZUFUEGEN (добавить) вызывает добавление в основной массив одной или нескольких записей в порядке сортировки основного массива. Формат инструкции:

ZUFUEGEN__;

Чтобы расположить правильно добавляемые записи, массивы с последовательной организацией переносятся на новый носитель.

Инструкция LOESCHEN (стереть) вызывает логическое стирание в основном массиве одной или нескольких записей. Формат инструкции:

LOESCHEN__;

Во входных записях для стирания достаточно иметь лишь идентификационные признаки стираемых записей.

Инструкция **STILLEGEN** (консервировать) вызывает исключение основных записей из общей обработки. Законсервированные записи остаются в массиве. Формат инструкции:

STILLEGEN__;

Инструкция **AKTIVIEREN** (активизировать) представляет законсервированные записи для дальнейшей обработки. Входные записи должны быть расположены в порядке их возрастания. Формат инструкции:

AKTIVIEREN__;

С помощью инструкции **AENDERN** (изменить) можно за один прогон программы выполнить различные функции по обслуживанию основного массива. По информации в поле функции входной записи определяется специальная функция для той или иной записи. Формат инструкции:

AENDERN__;

За один прогон программы одновременно могут быть выполнены следующие операции по обслуживанию массива: заменить, добавить, стереть, консервировать, активизировать, корректировать. Для выполнения этих операций в поле функции нужно занести букву или цифру, обозначающую соответствующую операцию:

- L или 1 — стереть;
- Z или 2 — добавить;
- V или 3 — корректировать;
- A или 6 — активизировать;
- S или 7 — консервировать;
- E или 8 — заменить.

Для записей сложной структуры знак или буква требуются только для первой физической записи. Входные записи для изменений должны быть рассортированы в порядке возрастания идентификационных признаков. Сортировка по признаку функции не требуется.

С помощью инструкции **ZUORDNEN** (подчинить) можно подчинить (явно или неявно) друг другу поля основного и входного массивов с помощью их имен. Неявное подчинение можно представить следующим образом:

ZUORDNEN [IMPL [{ NUR
 AUSSER }
 имя-поля 1,
 имя поля 2 ...]];

где IMPL — фиксированное слово, которое указывает, что поля основного массива должны быть связаны с полями входного массива неявно. При этом основные поля образуются из входных полей того же имени;

$\left\{ \begin{array}{l} \text{NUR} \\ \text{AUSSER} \end{array} \right\}$ — фиксированные слова, указывающие, что следующие за ними имена полей нужно исключать из неявного подчинения (AUSSER) или что следующие за ними имена полей должны быть подчинены друг другу (NUR). Можно задавать любое количество полей, разделенных запятой.

Явное подчинение можно представить следующим образом:

ZUORDNEN EXPL $\left[\text{UND IMPL} \left\{ \left\{ \begin{array}{l} \text{NUR} \\ \text{AUSSER} \end{array} \right\} \text{имя поля 1, имя поля 2, ...} \right\} \right]$;

Имя поля основного массива AUS имя поля входного массива;

где EXPL — фиксированное слово, указывающее, что поля основного и входного массивов, названные в последующих инструкциях AUS, должны быть явно подчинены друг другу;

UND IMPL — фиксированные слова, указывающие, что наряду с явным подчинением может выполняться и неявное. При этом все поля основного массива образуются из одноименных полей входного массива, если ни одно из них или оба поля не были подчинены явно или были исключены из подчинения словом AUSSER.

Имя поля основного массива должно начинаться в колонке 1;

AUS — фиксированное слово, которое соединяет подчиняемые друг другу поля основного и входного массивов.

С помощью инструкции ZUORDNEN (подчинить) подчиняются: имена информационных полей и соответствующие синонимы; идентификационные поля записей и их синонимы; имена текстовых частей и их синонимы. Не подчиняются: поле функции; поле состояния; идентификационное поле сегмента; поле последовательности.

С помощью инструкции EINFUEGEN (включить) устанавливается точка выхода для дополнительных модулей и одновременно сообщаются необходимые имена полей и данные пользователя. Формат инструкции:

EINFUEGEN имя дополнительного модуля

BEI

точка выхода

$\left[\text{MIT} \left\{ \begin{array}{l} \text{имя поля 1 [, имя поля 2, ...]} \\ \text{'данные пользователя'} \\ \text{'данные пользователя'} \end{array} \right\} \right];$

где дополнительный модуль должен быть самоперемещающимся и каталогизироваться в библиотеку абсолютных модулей (CL) в виде фазы;

BEI — фиксированное слово, соединяющее операнды «имя модуля» и «точка выхода» друг с другом. При накоплении предусмотрены следующие точки выхода: VOREIN, MODEIN; VORAUS. В одной программе в точках выхода должно быть подключено не более одного дополнительного модуля. Точки VOREIN и MODEIN не могут загружаться одновременно; при использовании этих точек нельзя указывать имена полей;

MIT — фиксированное слово, соединяющее точки выхода с последующими именами полей или данными пользователя. Общая длина всех имен полей не должна превышать 255 знаков; данные пользователя записываются в виде числовых констант и являются специальной информацией, необходимой только для дополнительного модуля. Их максимальная длина — не более 255 знаков.

Пример использования инструкции:

```
EINFUEGEN PHASE21 BEI MODEIN MIT  
FELD1, FELD2, AD212 ;
```

Параметры полей FELD1 и FELD2 выдаются пользователю в списке параметров. Адреса и длина данных пользователя AD 212 задаются в управляющем блоке дополнительного модуля.

8. ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ СИСТЕМЫ САВИ

Вспомогательные функции дают возможность пользователю получить сведения о содержании ТОД и обслуживать ее масса. Обеспечивают стирание ненужной ТОД и уплотнение массива.

Управляющая инструкция HILFSFUNKTION определяет выполнение последующих инструкций, относящихся к массиву ТОД или к таблице описаний данных. Формат инструкции:

```
HILFSFUNKTION {имя ТОД { [код доступа  
                        [код накопления ] ] } };
```

где DBTDATEI — фиксированное слово, указывающее, что последующие вспомогательные функции относятся к массиву ТОД.

Инструкция AUFLISTEN обеспечивает распечатку содержимого таблицы описания данных или массива ТОД. Формат инструкции:

```
AUFLISTEN_;
```

С помощью инструкции DATEIPARAMETER производится распечатка служебной информации о массиве ТОД, которая содержится в головной записи таблицы описания данных. Формат инструкции:

```
DATEIPARAMETER_;
```

Инструкция FELDPARAMETER дает возможность получить распечатку служебной информации о полях, которая содержится в таблице описания данных. Формат инструкции:

```
FELDPARAMETER { (NAMEN  
                  { ZAEHLER  
                  [ ATTRIBUTE ] } ) };
```

Благодаря инструкции LOESCHEN таблица описания данных, имя которой совпадает с именем, указанным в инструкции HILFSFUNKTION, логически стирается в массиве ТОД. Формат инструкции:

LOESCHEN__;

С помощью инструкции VERDICHTEN логически стертые таблицы описания данных физически удаляются из массива ТОД, т. е. массив ТОД реорганизуется таким образом, что все имеющиеся описания данных располагаются непосредственно друг за другом. Формат инструкции:

VERDICHTEN__;

Для работы программы с инструкциями вспомогательных функций указывается следующая последовательность управляющих карт задания:

```
// JOB имя задания
// ASSIGN SYS004,X'ppp' } адрес дискового устройства с мас-
// DLBL SWDBT, 'имя массива ТОД',DA) сивом ТОД
// EXTENT SYS004,...
// ASSIGN SYS005,X'ppp' } адрес дискового устройства с рабо-
// DLBL SWARB,,Z,DA ) чим массивом (для уплотнения)
// EXTENT SYS005,...
// EXEC SW#SAWI } инструкции для вспомогательных
: функций
:
/*
/&
```

В зависимости от операнда, заданного в инструкции, возможны следующие последовательности инструкций вспомогательных функций:

а) HILFSFUNKTION DBTDATEI;
AUFLISTEN;

или

VERDICHTEN;

Все другие инструкции игнорируются.

б) HILFSFUNKTION имя ТОД ...;

AUFLISTEN;
DATEIPARAMETER;
FELDPARAMETER...;
FELDPARAMETER...;

:

FELDPARAMETER...;

в) HILFSFUNKTION имя ТОД...;

LOESCHEN__;

г) HILFSFUNKTION имя ТОД...;

AUFLISTEN__;

Здесь система выполняет дополнительную работу, не заданную инструкцией DATEIPARAMETER.

9. ЯЗЫК ОПЕРАЦИЙ ДЛЯ ПОИСКА И ВЫВОДА ИНФОРМАЦИИ

Поиск в основных массивах осуществляется с помощью управляющей инструкции SUCHEN (поиск). Она вызывает программы поиска и подготовки ТОД для основного массива, в котором осуществляется поиск. Формат инструкции:

SUCHEN имя ТОД оси;

С помощью инструкции WENN (если) формулируются условия поиска. Формат инструкции:

```
[имя условия] WENN      операнд условия
                           [NICHT]
                           [оператор условия]
                           [операнд сравнения]
                           [{ UND
                           { ODER }
                           ... ... ];
```

где имя условия — 1—7 алфавитно-цифровых знаков. В одной программе возможно максимально 25 имен условия.

Операндом условия может быть: имя поля; имя условия, в круглых скобках; ключевой операнд (максимальная длина ключа 128 байт); код доступа пользователя;

операторами условия могут быть:

GLEICH —	равно
GROESSER —	больше
KLEINER —	меньше
BEREICH —	область
TABELLE —	таблица
DURCHSUCHEN —	просмотреть
ERFUELLT —	условие выполнено;

операндом сравнения может быть: имя поля; константа (ее максимальная длина 255 байт); таблица констант (ее максимальная длина 255 байт); цифровая величина; таблица цифровых величин.

С помощью инструкций AUSGEBEN (вывод), DRUCKEN (печатать) и STANZEN (перфорировать) устанавливается выходной носитель данных или указывается, следует игнорировать вывод или нет. Формат инструкции:

```
[имя вывода] { AUSGEBEN } { [ALLE]
                { DRUCKEN } { UNTERDRUCKEN }
                { STANZEN } { AUF носитель } };
```

В качестве носителя могут использоваться: диск (PLATTE), магнитная лента без метки (BAND), магнитная лента с метками (KBAND), АЦПУ (PARALLELDPRUCKER, исключая STANZEN), перфокарты (KARTE, исключая DRUCKEN), перфолента (LOCH-BAND). Если не задается носитель для вывода, то происходит вывод на устройства системного вывода.

Инструкция AUFBEREITEN (подготовить) указывает, что выходные записи должны быть подготовлены в соответствии с последующими инструкциями ZEILE (строка). Одновременно устанавливаются определенные параметры вывода. Формат инструкции:

```
AUFBEREITEN { SATZLAENGE   nnnn }
              { BLOCKFAKTOR nnn  }
              { ANZAHL      nnn  } ;
```

где операнд SATZLAENGE указывает максимальную длину подготовленной записи (десятичное число);

операнд BLOCKFAKTOR определяет фактор блокировки для выходного массива. Если фактор блокировки не указывается, то производится неблокированный вывод;

операнд ANZAHL определяет количество строк при выводе на печать в расчете на каждый лист. Если операнд не указывается, то производится вывод из расчета 60 строк на лист.

С помощью инструкции ZAEHLEN (считать) подсчитывается количество всех записей выходного массива. Формат инструкции:

ZAEHLEN _;

Если необходимо только определить общее количество записей выходного массива, то требуется следующая последовательность инструкций;

AUSGEBEN UNTERDRUECKEN; ZAEHLEN;

С помощью инструкции ADDIEREN (сложить) суммируются реквизиты цифрового поля по всем записям выходного массива; можно производить суммирование нескольких полей в одно поле результата. Формат инструкции:

ADDIEREN имя поля IN имя поля результата;

Инструкция SORTIEREN (сортировать) влияет на сортировку итоговых записей максимально по восьми ведущим реквизитам, по которым осуществляется сортировка. При этом длина всех ведущих реквизитов не должна превышать 256 байт. Инструкция действительна для любого выходного массива. Формат инструкции:

SORTIEREN имя поля { { STEIGEND }
 { FALLEND } }
 [имя поля { { STEIGEND }
 { FALLEND } } . . .] ;

где операнды { STEIGEND } (возрастающая) указывают на последовательность сортировки. Если операнды опущены, то предполагается STEIGEND.

Если операнды опущены, то предполагается STEIGEND.

Для выполнения сортировки требуется МЛ без метки для промежуточного накопления итоговых записей и рабочая область на МД (не менее 300 дорожек). Логическим устройствам SYS001 и SYS002 назначают одну МЛ, а затем применяют управляющий оператор //MTC REW, SYS002. Для распределения области на МД используется логическое имя массива SORTWK1 и логическое устройство SYS003.

Объем основной памяти при сортировке прежде всего зависит от длины итоговой записи. Учитывая, что система работает с перекрытием, объем основной памяти рассчитывается по следующей эмпирической формуле:

объем основной памяти = 2K + длина блока +
длина записи + объем основной памяти (SORT)

Инструкция TITEL (заголовок) описывает структуру строк титульного листа. Формат инструкции:

$$\text{TITEL} \left[\begin{array}{l} \text{'text'}_1' \\ \text{'text'}_2' \\ \text{'text'}_3' \\ \text{'text'}_4' \end{array} \begin{array}{l} \text{[nnn}_1\text{]} \\ \text{[nnn}_2\text{]} \\ \text{[nnn}_3\text{]} \\ \text{[nnn}_4\text{]} \end{array} \right];$$

где 'text' — указание текста, оно состоит не более чем из 120 или 156 знаков в зависимости от типа АЦПУ;

nnn — номер позиции начала текста в строке заглавия.

Инструкция KOPF (шапка) описывает структуру строк подзаголовка. Формат инструкции:

$$\text{KOPF} \left[\begin{array}{l} \text{'text'}_1' \\ \text{'text'}_2' \\ \text{'text'}_3' \\ \vdots \\ \text{'text'}_n' \end{array} \begin{array}{l} \text{[nnn}_1\text{]} \\ \text{[nnn}_2\text{]} \\ \text{[nnn}_3\text{]} \\ \vdots \\ \text{[nnn}_n\text{]} \end{array} \right];$$

где 'text' — указание текста, оно состоит не более чем из 120 или 156 знаков в зависимости от типа АЦПУ;

nnn — номер позиции начала текста в строке подзаголовка (шапки).

Инструкция FUSS (последняя строка) описывает структуру последней строки при выводе на печать. Последние строки распечатываются в конце каждого листа. Формат инструкции:

$$\text{FUSS} \left[\begin{array}{ll} \text{'text'}_1' & \text{nnn}_1 \\ \text{'text'}_2' & \text{nnn}_2 \\ \dots & \dots \end{array} \right];$$

где 'text' — текст, включающий не более чем 120 или 156 знаков в зависимости от типа АЦПУ;

nnn — номер позиции начала текста в заключительной строке.

С помощью инструкции VORSCHUB (сдвиг) для наглядного оформления табуляграммы могут быть введены пустые строки. В системе автоматический сдвиг на новую строку не предусмотрен. Формат инструкции:

$$\text{VORSCHUB } \text{nn};$$

где nn — десятичное число (максимум 30), указывающее, сколько пустых строк следует ввести между строками с информацией.

С помощью инструкции ZEILE (строка) запись результатов подготавливается для выдачи на печать. При этом пользователь устанавливает позицию вывода, форму представления и длину записи. Формат инструкции:

ZEILE	[{ имя поля 1 имя поля 1, данные пользователя 1 имя поля, 'шаблон']	
		[позиция]		
		{ { EXTERN INTERN HEXADEZIMAL DEZIMAL } }		;
		{ имя поля 2 имя поля 2, данные пользователя 2 'текст']	

где имя поля — имя основного поля, содержание которого должно выдаваться на печать в определенной форме;

имя поля, данные пользователя — имя основного поля, содержание которого должно выдаваться в той форме и с той длиной, которые устанавливаются пользователем.

В качестве данных пользователя в соответствии с формой представления основного поля могут быть использованы: $ppp1$ [, $ppp2$] (здесь $ppp1$ — десятичное число, указывающее количество выдаваемых знаков; $ppp2$ — десятичное число, указывающее относительный адрес выдаваемой части основного поля, относительно начала поля, имеющего форму представления T). Второе десятичное число может не указываться, если нужно вызвать первую часть поля (относительный адрес 0);

'шаблон' — операнд, который дает возможность определить шаблон для подготовки выдаваемого поля. Шаблон указывается для вывода во внешней форме (EXTERN) основного поля с формой представления A или P. Максимальная длина шаблона 31 знак (см. операнд MASKE). Сумма всех замененных цифрами знаков должна соответствовать разрядности поля и не должна превышать 15 байт. Если шаблон не указан ни в ТОД, ни в данной инструкции, система выдает содержимое поля в виде:

'????.????.????.???@?'

позиция — десятичное число, указывающее на позицию в строке для выдаваемого поля. Начальная позиция — единица. Если позиция не задана, то система разделяет в строке содержимое полей двумя пробелами;

операнд EXTERN (внешний) задает выдачу содержимого поля во внешней форме (EBCDIC) согласно ТОД;

операнд INTERN (внутренний) задает выдачу содержимого поля согласно форме представления, указанной в основной ТОД;

операнд HEXADEZIMAL (шестнадцатеричный) задает выдачу содержимого поля в шестнадцатеричной форме (X). При этом каждый байт представляется двумя знаками;

операнд DEZIMAL (десятичный) задает выдачу содержимого числового поля в десятичной форме.

С помощью инструкции ZEILE система определяет длину выходной записи, поэтому при выводе на ПК ее длина не должна превышать 80 знаков, а при выводе на печать — 120 или 156 знаков (в зависимости от типа АЦПУ).

Примеры использования ZEILE:

а) ZEILE FELD1 10 FELD2 35 TEXT, 30 65;
VORSCHUB 1;
ZEILE TEXT, 40, 30 65;

Здесь предусмотрена выдача записи в две строки, так как ширина печатной строки — 120 знаков;

б) ZEILE FELD1 15 FELD2, '?????,??' 30 'M' 48;

Здесь предусмотрена выдача записи, где заданы шаблон выдачи поля FELD2 и текст к данному полю.

ЛИТЕРАТУРА

1. Абдулаев А. А., Алиев Р. А., Уланов Г. М. Принципы построения автоматизированных систем управления промышленными предприятиями. М., «Энергия», 1975.
2. Брусенков И. В. [и др.]. Программная система управления базами данных СИНБАД. — В сб.: Алгоритмы и организация решения экономических задач. Вып. 6. М., «Статистика», 1975.
3. Велесько Е. И., Орлова Н. М., Пашкевич Б. В. Плановые расчеты в условиях АСУ. М., «Экономика», 1972.
4. Глушков В. М. Основные принципы построения автоматизированных систем управления. Киев, Укр.НИИНТИ, 1969.
5. Дудкина Л. В. [и др.]. Система управления базой данных «Банк». — В сб.: Алгоритмы и организация решения экономических задач. Вып. 8. М., «Статистика», 1976.
6. Козлова О. В., Кузнецов И. Н., Королев М. А. Научные основы управления производством. М., «Экономика», 1970.
7. Келехсаев А. А., Беляев А. П. Система интеграции и обработки данных СИОД1, СИОД2. М., «Статистика», 1977.
8. Проскуров В. С. Информация в АСПР. М., «Экономика», 1975.
9. Типовые проектные решения АСУП. Подсистема управления технической подготовкой производства. М., «Статистика», 1975.
10. Walter Beutel, Rolf Heinemann, Wolfgang Herzog, Peter Krah и. а. Datenspeicherung. Berlin. Verlag «Die Wirtschaft», 1974.

ОГЛАВЛЕНИЕ

Предисловие	3
Глава 1. ОРГАНИЗАЦИЯ И ВЕДЕНИЕ БАНКОВ ДАННЫХ В АСУП	
§ 1. Роль и место банка данных в АСУП	5
§ 2. Назначение и область применения системы БАСТАИ	9
§ 3. Состав и структура массивов системы БАСТАИ	11
§ 4. Создание программ организации и обслуживания банка данных	16
§ 5. Подготовка исходной информации для загрузки и изменения банка данных	20
§ 6. Макрокоманды системы БАСТАИ	23
§ 7. Программирование специальных программ пользователя (СПП)	30
Глава 2. ОРГАНИЗАЦИЯ И ВЕДЕНИЕ НЕЗАВИСИМЫХ МАССИВОВ ИНФОРМАЦИИ	
§ 1. Назначение системы программирования САВИ	38
§ 2. Создание таблиц описания данных	40
§ 3. Создание и ведение массивов	43
§ 4. Использование массивов	46
§ 5. Подготовка исходных данных	47
Глава 3. ОСОБЕННОСТИ РЕШЕНИЯ ЗАДАЧ В УСЛОВИЯХ СУЩЕСТВОВАНИЯ БАНКА ДАННЫХ	
§ 1. Описание фонда нормативно-справочной информации АСУ-1-МЧЗ	50
§ 2. Состав и структура массивов банка данных АСУ-1-МЧЗ	62
§ 3. Технология формирования базы данных (на примере АСУ-1-МЧЗ)	69
§ 4. Расчет применяемости деталей и сборочных единиц в изделии	92
§ 5. Расчет специфицированных и сводных норм расхода материалов	97
§ 6. Расчет сводных трудовых нормативов	102
§ 7. Расчет нормативных калькуляций	109
§ 8. Особенности проектирования информационной базы АСУП с использованием СУБД	120

ПРИЛОЖЕНИЯ.

1. Структура системной части записей массивов банка данных . . .	128
2. Параметры для генерации модулей программ создания и обслуживания банка данных	130
3. Управляющие карты для генерации модулей	153
4. Управляющие карты для изменения банка данных	153
5. Перечень элементов языка процедур системы САВИ	154
6. Язык описания данных для создания таблиц описания данных (ТОД)	156
7. Язык операций для создания и ведения основных массивов . . .	162
8. Вспомогательные функции системы САВИ	166
9. Язык операций для поиска и вывода информации	167
Литература	173

Создание и использование банков данных в АСУП

Рецензент *В. И. Подольский*

Редактор *Л. А. Табакова* Мл. редактор *Г. В. Розанова*

Техн. редактор *В. А. Чуракова*

Корректоры *Г. А. Башарина, З. С. Кандыба*

Худ. редактор *Н. А. Володина* Обложка художника *Т. Н. Погореловой*

ИБ № 237

Сдано в набор 10/V 1977 г. Подписано к печати 27/X 1977 г.

Формат бумаги 60×90¹/₁₆. Бумага № 2. Объем 11 печ. л. Уч.-изд. л. 11,14.

Усл. п. л. 11,0. Тираж 20 000 экз. А14419. (Тематич. план 1977 г. № 99).

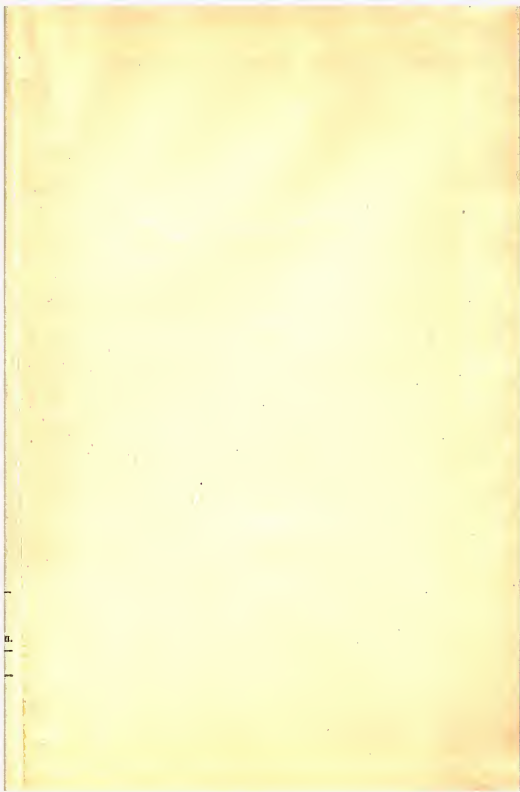
Заказ № 4038

Цена 65 коп.

Издательство «Статистика», Москва, ул. Кирова, 39.

Областная типография управления издательств,
полиграфии и книжной торговли

Ивановского облисполкома,
г. Иваново-8, ул. Типографская, 6.







65 коп.

THE NEW YORK PUBLIC LIBRARY ASTOR LENOX TILDEN FOUNDATION 500 5TH AVENUE NEW YORK 17, N.Y.